Private Information Retrieval from Transversal Designs

Julien Lavauzelle

Private Information Retrieval from Transversal Designs

Julien LAVAUZELLE
Laboratoire LIX, École Polytechnique, Inria & CNRS UMR 7161
Université Paris-Saclay

***Abstract*—Private information retrieval (PIR) protocols allow a user to retrieve entries of a database without revealing the index of the desired item. Information-theoretical privacy can be achieved by the use of several servers and specific retrieval algorithms. Most known PIR protocols focus on decreasing the number of bits exchanged between the client and the server(s) during the retrieval process. On another side, Fazeli *et al.* introduced so-called *PIR codes* in order to reduce the storage overhead on the servers. However, few works address the issue of the computation complexity of the servers.**

It this paper, we show that a specific encoding of the database yields PIR protocols with reasonable communication complexity, low storage overhead and optimal computational complexity for the servers. This encoding is based on incidence matrices of transversal designs, from which a natural and efficient recovering algorithm is derived. We also present several instances for our construction, which make use of finite geometries and orthogonal arrays. We finally give a generalisation of our main construction in order to resist collusions of servers.

I. INTRODUCTION

A. Private Information Retrieval

A private information retrieval (PIR) protocol aims at ensuring a user that he can retrieve some part D_i of a remote database D without revealing the index i to the server(s) holding the database. For example, such protocols can be applied in medical data storage where physicians would be able to access parts of the genome while hiding the specific gene they analyse. The PIR paradigm was originally introduced by Chor, Goldreich, Kushilevitz and Sudan [6, 7].

A naive solution to the problem consists in downloading the entire database each time the user wants a single entry. But the communication complexity would then be overwhelming, so we look for PIR protocols exchanging less bits. However, Chor *et al.* proved that, when the k-bits database is stored on a single server, a PIR protocol which leaks no information on the index i (such a protocol being called *information-theoretically secure*) must use $\Omega(k)$ bits of communication [7]. Two alternatives were then considered: restricting the protocol to computational security (initiated by Chor and Gilboa [5]), or allowing several servers to store the database. Our work focuses on the last one.

This paper appears in: IEEE Transactions on Information Theory, on pages: 1-17, DOI: 10.1109/TIT.2018.2861747.

It was presented in part at the Tenth International Workshop on Coding and Cryptography 2017, September 18-22, 2017, Saint-Petersburg, Russia.

This work is partially funded by French ANR-15-CE39-0013-01 "Manta".

In many such PIR protocols the database is *replicated* on ℓ servers, $\ell > 1$. Informally, the idea is that each server is asked to compute some partial information related to a random-like query sent by the user. Then the user collects all the servers' answers and retrieves the desired symbol with an appropriate algorithm. For instance, Chor *et al.* [7] considered a smart arrangement of the database entries in a $\log(\ell)$-dimensional array, and used XOR properties to mask the index of the desired item and to retrieve the associated symbol. Their protocol features decreasing communication as a function of the number of servers: with ℓ servers, the communication is $\mathcal{O}(\ell \log(\ell) k^{1/\log \ell})$ bits. For constant ℓ, the authors also proposed a PIR protocol with communication $\mathcal{O}(k^{1/\ell})$. A few years later, Katz and Trevisan [13] showed that any smooth locally decodable code $\mathcal{C} \subseteq \Sigma^n$ of locality ℓ gives rise to a PIR protocol with ℓ servers whose communication complexity is $\mathcal{O}(\ell \log(n|\Sigma|))$ — see [20] for a good survey on locally decodable codes (LDCs) and their applications in PIR protocols. Building on this idea, many PIR schemes (notably [3, 19, 10, 9]) successively decreased the communication complexity, achieving $\mathcal{O}(k^{\sqrt{\log\log k/\log k}})$ with only $\ell = 2$ servers. However, only few of them tried to lighten the computational and storage cost on the server side.

By preprocessing the database, Beimel, Ishai and Malkin [4] were the first to address the minimization of the server storage/computation in PIR protocols. Then, initiated by Fazeli, Vardy and Yaakobi [11], recent works used the concept of *PIR codes* to address the storage issue. The idea is to turn an ℓ-server replication-based PIR protocol into a more-than-ℓ-server distributed PIR protocol with a smaller overall storage overhead. For this purpose, the user encodes the database and distributes pieces of the associated codeword among the servers, such that servers hold distinct parts of the database (plus some redundancy). Through this transformation, both communication complexity and computational cost keep the same order of magnitude, but the storage overhead corresponds to the PIR code's one, which can be brought arbitrarily close to 1 when sufficiently many servers are used. Several recent works also address the PIR issue on previously coded databases [18], and/or aim at reaching the so-called capacity of the model [17]. However, while the storage drawback seems to be solved, huge computational costs still represent a barrier to the practicality of such PIR protocols.

B. Motivations and results

As pointed out by Yekhanin [20], "the overwhelming computational complexity of PIR schemes (...) currently presents the main bottleneck to their practical deployment". Consider a public database which is frequently queried, *e.g.* a database storing stock exchange prices where private queries could be very relevant. Fast retrieval is crucial is this context. Hence, one cannot afford each run of the PIR protocol to be computationally inefficient, for instance $\Omega(k)$ if k is the size of the database. Therefore, a relevant goal is to build PIR protocols with sublinear computational complexity in the length of the database stored by each server.

Naively, the computational complexity of a PIR protocol could be drastically reduced if we let all possible answers to its queries to be precomputed. Of course, storing all these answers dramatically increases the needed storage, so let us focus on a construction due to Augot, Levy-dit-Vehel and Shikfa [2] — anterior to the PIR codes breakthrough [11] — that address this issue.

The construction of Augot *et al.* [2] uses a specific family of high-rate locally decodable codes called *multiplicity codes* introduced by Kopparty, Saraf and Yekhanin [14]. But instead of *replicating* the database on ℓ servers ($\ell > 1$ being the locality of the codes), the authors *split* an encoded version c of the database D into parts $c^{(1)}, \ldots, c^{(\ell)}$, and share these parts on the servers. The main difference with PIR codes [11] is that Augot *et al.*'s construction does not purpose to *emulate* a lighter PIR protocol with an existing one. It uses specific properties of the encoding as a way to split the database on several servers. In short, the multiplicity codes they use feature *both* the privacy of the PIR protocol and the storage reduction for the servers. We refer to Section VII for more details on the construction.

In this work, we reconsider this "codeword support splitting" idea, and we propose a new generic framework for the construction of PIR protocols which takes into account the computational complexity issue. More precisely, the protocols we give are computationally optimal with respect to the communication complexity of the protocol, in the sense that each server needs to read *only one* entry in the part of the database it holds.

Our construction is based on combinatorial structures called *transversal designs*, from which we naturally derive a linear code, a partition of its support and a *local* reconstruction algorithm. In practice, we give several instances of transversal designs that lead to codes with large rate, hence to PIR protocols with low storage overhead. The two first families come from incidences between points and lines in the affine (resp. projective) space. They are closely related to the classical geometric designs of 1-flats. A third family of instances makes use of a classical transformation of so-called *orthogonal arrays* of strength 2 into transversal designs. We then proceed to a thorough study of the dimension of codes coming from *MDS-like* orthogonal arrays of strength 2. A fourth and last family of practical instances appears when showing that orthogonal arrays built from *divisible codes* lead to PIR protocols with storage expansion less than 2. We finally prove that orthogonal arrays with strength $t > 2$ allow the construction of PIR protocols resisting to collusions of up to $t - 1$ servers. We exhibit and analyzed instances of some orthogonal arrays with large strength to conclude this work.

C. Organization

We start by giving two formal definitions of PIR protocols in Section II, depending on whether the database is replicated or distributed on the servers. We also present the standard construction of replication-based PIR protocols from smooth locally decodable codes. In Section III, we recall definitions of combinatorial structures and their associated codes. The 1-private PIR protocols based on transversal designs are introduced in Section IV. Section V is devoted to four families of instances of the PIR construction having practical parameters. Finally, a generalisation of our construction is given in Section VI in order to keep up with collusions of servers, and a comparison with the PIR protocols coming from multiplicity codes is presented in Section VII.

II. Definitions and related constructions

We first recall that we are only concerned with information-theoretically secure PIR protocols. In this paper, we denote by U the *user* (or *client*) of the PIR protocol. User U owns a database denoted by $D = (D_i)_{1 \leq i \leq k} \in \mathbb{F}_q^k$, where $\mathbb{F}_q$ represents the finite field with q elements. Database D hence contains $|D| = k \log q$ bits. We also denote by $S_1, \ldots, S_\ell$ the ℓ servers involved in the PIR protocol.

Given A, B two sets, with $|B| = n < \infty$, we denote by A^B the set of n-tuples $a = (a_b)_{b \in B}$ of A-elements indexed by B, which can also be seen as functions from B to A. For $T \subset B$, we also write $a_{|T} \coloneqq (a_t)_{t \in T}$ the restriction of the tuple a to the coordinates of T.

A. Two definitions for PIR protocols

A vast majority of existing PIR schemes start by simply cloning the database D on all the servers $S_1, \ldots, S_\ell$. Then, the role of each server S_j is to compute some combination of symbols from D, related to the query sent by U. This computation has a non-trivial cost, so in a certain sense, the computational complexity of the privacy of the PIR scheme is mainly devoted to the servers.

More formally, one can define *replication-based PIR protocols* as follows:

Definition II.1 (standard, or replication-based PIR protocol)**.** Assume that every server S_j, $1 \leq j \leq \ell$, stores a copy of the database D. An ℓ-server replication-based PIR protocol is a set of three algorithms $(\mathcal{Q}, \mathcal{A}, \mathcal{R})$ running the following steps on input $i \in [1, k]$:

1) *Query generation:* the randomized algorithm $\mathcal{Q}$ generates ℓ queries $(q_1, \ldots, q_\ell) \coloneqq \mathcal{Q}(i)$. Query q_j is sent to server S_j.
2) *Servers' answer:* each server S_j computes an answer $a_j = \mathcal{A}(q_j, D)$ and sends it back to the user[1].

[1] algorithm $\mathcal{A} \coloneqq \mathcal{A}_j$ may depend on j

3) *Reconstruction:* denote by $\mathbf{a} = (a_1, \ldots, a_\ell)$ and $\mathbf{q} = (q_1, \ldots, q_\ell)$. User U computes and outputs $r = \mathcal{R}(i, \mathbf{a}, \mathbf{q})$.

The PIR protocol is said:

- *correct* if $r = D_i$ when the servers follow the protocol;
- *t-private* if, for every $(i, i') \in [1, k]^2$ and every $T \subseteq [1, \ell]$ such that $|T| \leq t$, the distributions $\mathcal{Q}(i)_{|T}$ and $\mathcal{Q}(i')_{|T}$ are the same. We also say that the PIR protocol resists t collusions of servers.

We call *communication complexity* the number of bits sent between the user and the servers, and *server* (resp. *user*) *computational complexity* the maximal number of $\mathbb{F}_q$-operations made by a server in order to compute an answer a_j (resp. made by $\mathcal{R}$ to reconstruct the desired item).

According to this definition, one sees that the servers must jointly carry the ℓ copies of the database, so the *storage overhead* of the scheme is $(\ell - 1)|D|$ bits. Moreover, since D is a raw database without specific structure, the algorithm $\mathcal{A}$ has no reason to be trivial and can incur superlinear computations for the servers — which is verified for most of current replication-based PIR protocols.

A way to reduce the computation cost of PIR protocols is to preprocess the database. Therefore we need to model PIR protocols for which the database can be encoded and distributed over the servers. From now on, let $c = (c_i)_{i \in I}$ denote *an encoding of the database* D, *i.e.* the image of D by an injective map $\mathbb{F}_q^k \to \mathbb{F}_q^I$, with $|I| = n \geq k$. Besides, for convenience we assume that $I = [1, s] \times [1, \ell]$ and for readability we write $c_{(i_1, i_2)} = c_{i_1}^{(i_2)}$ and $c^{(j)} = (c_r^{(j)})_{r \in [1,s]}$.

Definition II.2 (distributed PIR protocol)**.** Assume that for $1 \leq j \leq \ell$, server S_j holds the part $c^{(j)}$ of the encoded database. An ℓ–server distributed PIR protocol is a set of three algorithms $(\mathcal{Q}, \mathcal{A}, \mathcal{R})$ running the following steps on input $i \in I$:

1) *Query generation:* the randomized algorithm $\mathcal{Q}$ generates ℓ queries $(q_1, \ldots, q_\ell) \coloneqq \mathcal{Q}(i)$. Query q_j is sent to server S_j.
2) *Servers' answer:* each server S_j computes an answer $a_j = \mathcal{A}(q_j, c^{(j)})$ and sends it back to the user.
3) *Reconstruction:* denote by $\mathbf{a} = (a_1, \ldots, a_\ell)$ and $\mathbf{q} = (q_1, \ldots, q_\ell)$. User U computes and outputs $r = \mathcal{R}(i, \mathbf{a}, \mathbf{q})$.

Correctness and *privacy* properties are identical to those of replication-based PIR protocols. Similarly, one can also define *communication* and *computational complexities*, and since the database D has been encoded, we finally define the *storage overhead* as the number of redundancy bits stored by the servers, that is, $(s\ell - k) \log q$.

In this paper, we focus on distributed PIR protocols with low computational complexity on the server side. More precisely, we build PIR protocols where the answering algorithm $\mathcal{A}$ consists only in *reading some symbols* of the database. Thus, our PIR protocols are computationally optimal on the server side, in a sense that, compared to the non-private retrieval, they incur no extra computational burden for the each server taken individually.

B. PIR protocols from locally decodable codes

As pointed out in the introduction, Augot *et al.* [2] used a family of locally decodable codes (LDC) to design a distributed PIR scheme. LDCs are known to give rise to PIR protocols for a long time [13], but we emphasize that the main idea from [2] is to benefit from the fact that the encoded database can be smartly partitioned with respect to the queries of the local decoder.

Based on the seminal work of Katz and Trevisan [13], we briefly remind how to design a PIR protocol based on a perfectly smooth locally decodable code. First, let us define (linear) locally decodable codes.

Definition II.3 (locally decodable code)**.** Let Σ be a finite set, $2 \leq \ell \leq k \leq n$ be integers, and $\delta, \epsilon \in [0, 1]$. A code $\mathcal{C} : \Sigma^k \to \mathbb{F}_q^n$ is (ℓ, δ, ϵ)–locally decodable if and only if there exists a randomized algorithm $\mathcal{D}$ such that, for every input $i \in [1, k]$ we have:

- for all $m \in \Sigma^k$ and all $y \in \mathbb{F}_q^n$, if $|\{j \in [1, n], y_j \neq \mathcal{C}(m)_j\}| \leq \delta n$, then
$$\mathbb{P}(\mathcal{D}^{(y)}(i) = m_i) \geq 1 - \epsilon\,,$$
where the probability is taken over the internal randomness of $\mathcal{D}$;
- $\mathcal{D}$ reads at most ℓ symbols $y_{q_1}, \ldots, y_{q_\ell}$ of y.

Notation $\mathcal{D}^{(y)}$ refers to the fact that $\mathcal{D}$ has oracle access to single symbols y_{q_j} of the word y. The parameter ℓ is called the *locality* of the code. Moreover, the code $\mathcal{C}$ is said *perfectly smooth* if on an arbitrary input i, each individual query of the decoder $\mathcal{D}$ is uniformly distributed over the coordinates of the word y.

Now let us say a user wants to use a PIR protocol on a database $D \in \Sigma^k$, and assume there exists a perfectly smooth locally decodable code $\mathcal{C} \subset \mathbb{F}_q^n$ of dimension k and locality ℓ. Figure 1 presents a distributed PIR protocol based on $\mathcal{C}$.

1) Initialization step. User U encodes D into a codeword $c' \in \mathcal{C}$. Each server $S_1, \ldots, , S_\ell$ holds a copy of c'. In the formalism of Definition II.2, it means that $c^{(j)} \coloneqq c'$, for $j = 1, \ldots, \ell$.

2) Retrieving step for symbol D_i. Denote by $\mathcal{D}$ a local decoding algorithm for $\mathcal{C}$.

1) *Queries generation:* user U calls $\mathcal{D}$ to generate at random a query $(q_1, \ldots, q_\ell)$ for decoding the symbol D_i. Query q_j is sent to server S_j.
2) *Servers' answer:* each server S_j *reads* the encoded symbol $a_j \coloneqq c'_{q_j}$. Then S_j sends a_j to U.
3) *Reconstruction:* user U collects the ℓ codeword symbols $(c'_{q_j})_{j \in [1,\ell]}$ and feeds the local decoding algorithm $\mathcal{D}$ in order to retrieve D_i.

Fig. 1: A distributed PIR protocol based on a locally decodable code $\mathcal{C}$.

The main drawback of these LDC-based PIR protocols is their storage overhead, since the ℓ servers must store $\ell n/k = \ell/R$ times more data than the raw database ($R \coloneqq k/n$

represents the *information rate*, or *rate*, of the code). This issue becomes especially crucial as building LDCs with small locality and high rate is highly non-trivial.

The idea of Augot, Levy-dit-Vehel and Shikfa [2] for reducing the storage overhead is to benefit from a natural partition of the support of multiplicity codes [14]. Assume that each codeword $c \in \mathcal{C}$ can be *split* into ℓ disjoint parts $c^{(1)}, \ldots, c^{(\ell)}$, such that each coordinate q_j of any possible query $(q_1, \ldots, q_\ell)$ of the PIR protocol corresponds to reading some symbols on $c^{(j)}$. By sending the part $c^{(j)}$ to server S_j, the PIR protocol of Figure 1 can be improved in order to save storage. We devote Section VII to more explanation on this construction, as well as to a comparison with our schemes.

Finally, one can notice that the communication complexity of LDC-based PIR protocols depends on the locality of the code, while the smoothness of the code serves their privacy. We also point out two important remarks.

1) Assuming a noiseless transmission and *honest-but-curious* servers (*i.e.* they want to discover the index of the desired symbol but never give wrong answers), one *does not need* a powerful local decoding algorithm. Indeed, it should be possible to reconstruct the desired symbol D_i by local decoding only one erasure on the codeword. For instance, computing a single low-weight parity-check sum should be enough.
2) Smoothness is sufficient for 1-privacy, but we need more structure for preventing collusions of servers.

Coupled with the fact that we want to split the database over several servers, these remarks lead us to design other kinds of encoding, which answer as close as possible the needs of private information retrieval protocols. Our construction relies on combinatorial structures, namely *transversal designs*, that we recall in the upcoming section.

III. Transversal designs and codes

Let us give here the definition of transversal designs and how to build linear codes upon them. We refer to [1], [16] and [8] for complementary details.

Definition III.1 (block design)**.** A *block design* is a pair $\mathcal{D} = (X, \mathcal{B})$ where X is a finite set of so-called *points*, and $\mathcal{B}$ is a set of non-empty subsets of X called the *blocks*.

Definition III.2 (incidence matrix)**.** Let $\mathcal{D} = (X, \mathcal{B})$ be a block design. An *incidence matrix* $M_\mathcal{D}$ of $\mathcal{D}$ is a matrix of size $|\mathcal{B}| \times |X|$, whose $(i,j)-$entry, for $i \in \mathcal{B}$ and $j \in X$, is:

$$\begin{cases} 1 & \text{if the block } i \text{ contains the point } j, \\ 0 & \text{otherwise.} \end{cases}$$

The *q-rank* of $M_\mathcal{D}$ is the rank of $M_\mathcal{D}$ over the field $\mathbb{F}_q$.

For $B \subset X$, the *incidence vector* $\mathbb{1}_B \in \{0,1\}^X$ is the row vector whose x-th coordinate is 1 if and only if $x \in B$. Let us notice that, given a design $\mathcal{D} = (X, \mathcal{B})$, one can build $M_\mathcal{D}$ by stacking incidence vectors of blocks $B \in \mathcal{B}$.

Of course, any design admits many incidence matrices, depending on the way points and blocks are ordered. However, all these incidence matrices are equal up to some permutation of their rows and columns, and, in particular, they all have the same q-rank. Hence, we call *q-rank of a design* the q-rank of any of its incidence matrices. Moreover, from now on we consider incidence matrices of designs up to an ordering of points and blocks, and we abusively refer to *the* incidence matrix $M_\mathcal{D}$ of a design $\mathcal{D}$.

Example III.3. Let $\mathbb{A}^2(\mathbb{F}_3)$ be the affine plane over the finite field $\mathbb{F}_3$, and X be the set consisting of its 9 points:

$$X = \{\, (0,0), (0,1), (0,2), (1,0), (1,1), \\ (1,2), (2,0), (2,1), (2,2) \,\}.$$

We define the block set $\mathcal{B}$ as the set of the 12 affine lines of $\mathbb{A}^2(\mathbb{F}_3)$:

$$\begin{aligned} \mathcal{B} = \{\, &\{(0,0),(0,1),(0,2)\}, \{(1,0),(1,1),(1,2)\}, \\ &\{(2,0),(2,1),(2,2)\}, \{(0,0),(1,1),(2,2)\}, \\ &\{(1,0),(2,1),(0,2)\}, \{(2,0),(0,1),(1,2)\}, \\ &\{(0,0),(2,1),(1,2)\}, \{(1,0),(0,1),(2,2)\}, \\ &\{(2,0),(1,2),(0,2)\}, \{(0,0),(1,0),(2,0)\}, \\ &\{(0,1),(1,1),(2,1)\}, \{(0,2),(1,2),(2,2)\} \,\}. \end{aligned}$$

The pair $\mathcal{D} = (X, \mathcal{B})$ is then a block design, and its associated (12×9)–incidence matrix is

$$M_\mathcal{D} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

A computation shows that over the field $\mathbb{F}_2$, matrix $M_\mathcal{D}$ is full-rank, while over $\mathbb{F}_3$, it has only rank 6.

Definition III.4 (transversal design)**.** Let $s, \ell \geq 2$ and $\lambda \geq 1$ be integers. A *transversal design*, denoted $\mathrm{TD}_\lambda(\ell, s)$, is a block design $(X, \mathcal{B})$ equipped with a partition $\mathcal{G} = \{G_1, \ldots, G_\ell\}$ of X called the set of *groups*, such that:

- $|X| = \ell s$;
- any group in $\mathcal{G}$ has size s and any block in $\mathcal{B}$ has size ℓ;
- any unordered pair of elements from X is contained either in one group and no block or in no group and λ blocks.

If $\lambda = 1$, we use the simpler notation $\mathrm{TD}(\ell, s)$.

Remark III.5. A block cannot be secant to a group in more than one point, otherwise the third condition of the definition would be disproved. Moreover, since the block size equals the number of groups, any block must meet any group. Hence the following holds:

$$\forall (B, G) \in \mathcal{B} \times \mathcal{G},\ |B \cap G| = 1\,.$$

The definition also implies there must lie exactly λs^2 blocks in $\mathcal{B}$.

Example III.6. Let $\mathcal{D} = (X, \mathcal{B})$ be the block design defined in Example III.3. Define $\mathcal{G}$ to be any set of 3 parallel lines

from $\mathcal{B}$ which partitions the point set X. For instance, one can consider

$$\mathcal{G} = \{\, \{(0,0),(0,1),(0,2)\}, \\ \{(1,0),(1,1),(1,2)\}, \\ \{(2,0),(2,1),(2,2)\}\,\}\,.$$

Then, $\mathcal{T} = (X, \mathcal{B} \setminus \mathcal{G}, \mathcal{G})$ is a transversal design $\mathrm{TD}(3,3)$. Indeed, $\mathcal{T}$ is composed of $\ell s = 9$ points, $\ell = 3$ groups of size $s = 3$ and $s^2 = 9$ blocks of size $\ell = 3$ each. Moreover, in the affine plane every unordered pair of points belongs simultaneously to a unique line, which is represented in $\mathcal{T}$ either by a group or by a block. More generally, for any prime power q, a transversal design $\mathrm{TD}(q,q)$ can be built with the affine plane $\mathbb{A}^2(\mathbb{F}_q)$. A generalisation of this construction will be given in Subsection V-A.

A simple way to build linear codes from block designs is to associate a parity-check equation of the code to each incidence vector of a block of the design. We recall that the *dual code* $\mathcal{C}^\perp$ of a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is the linear vector space consisting of vectors $h \in \mathbb{F}_q^n$ such that $\forall c \in \mathcal{C}, \sum_{i=1}^n c_i h_i = 0$.

Definition III.7 (code of a design)**.** Let $\mathbb{F}_q$ be a finite field, $\mathcal{D} = (X, \mathcal{B})$ be a block design and $M_\mathcal{D}$ be its incidence matrix. The code $\mathrm{Code}_q(\mathcal{D})$ is the $\mathbb{F}_q$-linear code of length $|X|$ admitting $M_\mathcal{D}$ as a parity-check matrix.

Remark III.8. The code $\mathrm{Code}_q(\mathcal{D})$ is uniquely defined up to a chosen order of the points X. For different orders, the arising codes remain permutation-equivalent. Also notice that the way blocks are ordered does not affect the code.

For any design $\mathcal{D}$, the dimension over $\mathbb{F}_q$ of $\mathrm{Code}_q(\mathcal{D})$ equals $|X| - \mathrm{rank}_q(M_\mathcal{D})$. Since $M_\mathcal{D}$ has coefficients in $\{0,1\}$, one must notice that $\mathrm{rank}_q(M_\mathcal{D}) = \mathrm{rank}_p(M_\mathcal{D})$, where p is the characteristic of the field $\mathbb{F}_q$.

Remark III.9. Standard literature (*e.g.* [1]) sometimes defines $\mathrm{Code}_q(\mathcal{D})$ (and not $\mathrm{Code}_q(\mathcal{D})^\perp$) to be the vector space *generated* by the incidence matrix of the design. We favor this convention because $\mathrm{Code}_q(\mathcal{D})$ will serve to *encode* the database in our PIR scheme.

Example III.10. The design $\mathcal{D}$ from Example III.3 gives rise to $\mathcal{C} = \mathrm{Code}_3(\mathcal{D})$, a linear code over $\mathbb{F}_3$, of length 9 and dimension 3. A full-rank generator matrix of $\mathcal{C}$ is given by:

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \end{pmatrix}.$$

One may notice that this code is the generalized Reed-Muller code of degree 1 and order 2 over $\mathbb{F}_3$, that is, the evaluation code of bivariate polynomials of total degree at most 1 over the whole affine plane $\mathbb{F}_3^2$.

Definition III.11 (systematic encoding)**.** Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a linear code of dimension $k \leq n$. A systematic encoding for $\mathcal{C}$ is a one-to-one map $\phi : \mathbb{F}_q^k \to \mathcal{C}$, such that there exists an injective map $\sigma : [1,k] \to [1,n]$ satisfying:

$$\forall m \in \mathbb{F}_q^k, \forall i \in [1,k],\ m_i = \phi(m)_{\sigma(i)}\,.$$

The set $\sigma([1,k]) \subseteq [1,n]$ is called an *information set* of $\mathcal{C}$.

In other words, a systematic encoding allows to view the message m as a subword of its associated codeword $\phi(m) \in \mathcal{C}$. For instance, it is useful for retrieving m from c efficiently, when the codeword c has not been corrupted. A systematic encoding exists for any code $\mathcal{C}$, is not necessarily unique, and can be computed through a Gaussian elimination over any generator matrix of the code. Also notice that this computation can be tedious for large codes.

IV. 1-PRIVATE PIR PROTOCOLS BASED ON TRANSVERSAL DESIGNS

In this section we present our construction of PIR protocols relying on transversal designs. The idea is that the knowledge of one point of a block of a transversal design gives (almost) no information on the other points lying on this block. The code associated to such a design then transfers this property to the coordinates of codewords. Hence, we obtain a PIR protocol which can be proven 1-private, that is, which ensures perfect privacy for non-communicating servers. Though this protocol cannot resist collusions, we will see in Section VI that a natural generalisation leads to t-private PIR protocols with $t > 1$.

Notice that both Fazeli *et al.*'s work [11] and ours make use of codes in order to save storage in PIR protocols. Nevertheless, we emphasize that the constructions are very different, since Fazeli *et al.* emulate a PIR protocol from an existing one while we build our PIR protocols from scratch.

A. The transversal-design-based distributed PIR protocol

Let $\mathcal{T}$ be a transversal design $\mathrm{TD}(\ell, s)$ and $n = |X| = \ell s$. Denote by $\mathcal{C} = \mathrm{Code}_q(\mathcal{T}) \subseteq \mathbb{F}_q^n$ the associated $\mathbb{F}_q$-linear code, and let $k = \dim_{\mathbb{F}_q} \mathcal{C}$. Our PIR protocol is defined in Figure 2. We then summarize the steps of the construction in Figure 3.

B. Analysis

We analyse our PIR scheme by proving the following:

Theorem IV.1. *Let D be a database with k entries over $\mathbb{F}_q$, and $\mathcal{T} = \mathrm{TD}(\ell, s)$ be a transversal design, whose incidence matrix has rank $\ell s - k$ over $\mathbb{F}_q$. Then, there exists a distributed ℓ-server 1-private PIR protocol with:*

- *only one $\mathbb{F}_q$-symbol to read for each server,*
- *$\ell - 1$ field operations over $\mathbb{F}_q$ for the user,*
- *$\ell \log(sq)$ bits of communication ($\ell \log s$ are uploaded, $\ell \log q$ are downloaded),*
- *a (total) storage overhead of $(\ell s - k)\log q$ bits on the servers.*

Proof. Recall the PIR protocol we are dealing with is defined in Figure 2.

Correctness. By definition of the code $\mathcal{C} = \mathrm{Code}_q(\mathcal{T})$, the incidence vector $\mathbb{1}_B$ of any block $B \in \mathcal{B}$ belongs to the dual code $\mathcal{C}^\perp$. Hence, for $c \in \mathcal{C}$, the inner product $\mathbb{1}_B \cdot c$ vanishes, or said differently, $\sum_{x \in B} c_x = 0$. We recall that j^* represents the index of the group which contains i. Since the servers S_j, $j \neq j^*$, receive queries corresponding to the points of a block B which contains i, we have $c_i = -\sum_{x \in B \setminus \{i\}} c_x = -\sum_{j \neq j^*} c_{q_j}$, and our PIR protocol is correct as long as there is no error on the symbols $a_j \coloneqq c_{q_j}$ returned by the servers.

Parameters: $\mathcal{T} = (X, \mathcal{B}, \mathcal{G})$ is a $\mathrm{TD}_\lambda(\ell, s)$; $\mathcal{C} = \mathrm{Code}_q(\mathcal{T})$ has length $n = \ell s$ and dimension k.

1) Initialization step.

1) *Encoding.* User U computes a systematic encoding of the database $D \in \mathbb{F}_q^k$, resulting in the codeword $c \in \mathcal{C}$.
2) *Distribution.* Denote by $c^{(j)} = c_{|G_j}$ the symbols of c whose support is the group $G_j \in \mathcal{G}$. Each server S_j receives $c^{(j)}$, for $1 \leq j \leq \ell$.

2) Retrieving step for symbol c_i for $i \in X$. Denote by $j^* \in [1, \ell]$ the index of the unique group G_{j^*} which contains i — that is, $c_i = c_r^{(j^*)}$ for some $r \in [1, s]$. Also denote by $\mathcal{B}^*$ the subset of blocks containing i. The three steps of the distributed PIR protocol are:

1) *Queries generation.* U picks uniformly at random a block $B \in \mathcal{B}^*$. For $j \neq j^*$, user sends the unique index $q_j \in B \cap G_j$ to server S_j. Server S_{j^*} receives a random query q_{j^*} uniformly picked in G_{j^*}. To sum up ($\xleftarrow{\$}$ stands for "picked uniformly at random in"):

$$\begin{cases} \mathcal{Q}(i)_{j^*} \xleftarrow{\$} G_{j^*}, & \text{for } j^* \text{ s.t. } i \in G_{j^*} \\ B \xleftarrow{\$} \mathcal{B}^* & \\ \mathcal{Q}(i)_j \leftarrow B \cap G_j, & \text{for } j \neq j^* \end{cases}$$

2) *Servers' answer.* Each server S_j (including S_{j^*}) reads $a_j \coloneqq c_{q_j}$ and sends it back to the user. That is,

$$\mathcal{A}(q_j, c^{(j)}) = c_{q_j} .$$

3) *Reconstruction.* Denote by $\mathbf{a} = \{a_1, \ldots, a_\ell\}$ and $\mathbf{q} = \{q_1, \ldots, q_\ell\}$. User U computes

$$r = \mathcal{R}(i, \mathbf{a}, \mathbf{q}) \coloneqq -\sum_{j \neq j^*} a_j = -\sum_{j \neq j^*} c_{q_j}$$

and outputs r.

Fig. 2: A 1-private distributed PIR protocol based on the $\mathbb{F}_q$-linear code defined by a transversal design.

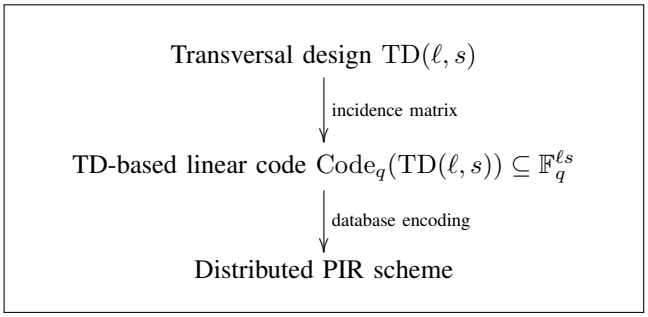

Fig. 3: Summary of the steps leading to the construction of a transversal-design-based PIR scheme.

Security (1-privacy). We need to prove that for all $j \in [1, \ell]$, it holds that $\mathbb{P}(i \mid q_j) = \mathbb{P}(i)$, where probabilities are taken over the randomness of $B \leftarrow \mathcal{B}^*$. The law of total probability implies

$$\begin{aligned} \mathbb{P}(i \mid q_j) &= \mathbb{P}(i \mid q_j \text{ and } i \in G_j)\, \mathbb{P}(i \in G_j) \\ &\quad + \mathbb{P}(i \mid q_j \text{ and } i \notin G_j)\, \mathbb{P}(i \notin G_j) \\ &= \mathbb{P}(i \mid i \in G_j)\, \mathbb{P}(i \in G_j) \\ &\quad + \mathbb{P}(i \mid i \notin G_j)\, \mathbb{P}(i \notin G_j) \\ &= \mathbb{P}(i) , \end{aligned}$$

and the reasons why we eliminated the random variable q_j in the conditional probabilities are:

- in the case $i \in G_j$ (that is, $j = j^*$), by definition of our PIR protocol we know that q_j is uniformly random, so q_j and i are independent;
- in the case $i \notin G_j$, by definition of a transversal design, there are as many blocks containing both q_j and i as there are blocks containing q_j and any i' in $X \setminus G_j$ (the number of such blocks is always λ). So once again, the value of the random variable q_j is not related to i.

Communication complexity. Exactly one index in $[1, s]$ and one symbol in $\mathbb{F}_q$ are exchanged between each server and the user. So the overall communication complexity is $\ell \times (\log(s) + \log(q)) = \ell \log(sq)$ bits.

Storage overhead. The number of bits stored on a server is $s \log q$, giving a total storage overhead of $(\ell s - k) \log q$, where $k = \dim \mathcal{C}$.

Computation complexity. Each server S_j only needs to read the symbol defined by query q_j, hence our protocol incurs no extra computational cost. □

Theorem IV.1 shows that, if we want to optimize the practical parameters of our PIR scheme, we basically need to look for small values of ℓ, the number of groups. However, one observes that the dimension k of $\mathrm{Code}_q(\mathcal{T})$ strongly depends on ℓ and n, and tiny values of ℓ can lead to trivial or very small codes. This issue should be carefully taken into account, since instances with $k < \ell$ represent PIR protocols which are more communication expensive to use than the trivial one, which simply retrieves the whole database. Hence, it is very natural to raise the main issue of our construction:

Problem IV.2. *Find codes $\mathcal{C} = \mathrm{Code}_q(\mathcal{T})$ arising from transversal designs $\mathcal{T} = \mathrm{TD}(\ell, s)$ with few groups (small ℓ) and large dimension $k = \dim_{\mathbb{F}_q} \mathcal{C}$ compared to their length $n = \ell s$.*

We first give a negative result, stating that the characteristic of the field $\mathbb{F}_q$ should be chosen very carefully in order to obtain non-trivial codes.

Proposition IV.3. *Let $\mathcal{T} = (X, \mathcal{B}, \mathcal{G})$ be a $\mathrm{TD}_\lambda(\ell, s)$. Let $q = p^e$, p prime. If $p \nmid \lambda s$, then*

$$\mathrm{Code}_q(\mathcal{T}) \subseteq \{c \in \mathbb{F}_q^{s\ell}, \forall G \in \mathcal{G}, c_{|G} \in \mathrm{Rep}(s)\} ,$$

where $\mathrm{Rep}(s)$ represents the repetition code of length s. In particular, if $p \nmid \lambda s$, then $\mathrm{Code}_q(\mathcal{T})$ has dimension at most ℓ.

Proof. For $x \in X$, recall that $\mathcal{B}_x = \{B \in \mathcal{B}, x \in B\}$, and denote by $a^{(x)} = \sum_{B \in \mathcal{B}_x} \mathbb{1}_B$. We know that $a^{(x)} \in \mathrm{Code}_q(\mathcal{T})^\perp$,

since $\mathrm{Code}_q(\mathcal{T})^\perp$ is generated by $\{\mathbb{1}_B, B \in \mathcal{B}\}$. Denote by $G_x \in \mathcal{G}$ the only group that contains x. We see that:

$$\begin{cases} a_x^{(x)} = \lambda s \\ a_i^{(x)} = 0 & \text{for all } i \in G_x \setminus \{x\} \\ a_j^{(x)} = \lambda & \text{for all } j \in X \setminus G_x\,. \end{cases}$$

Therefore $a^{(x)} - a^{(y)} = \lambda s(\mathbb{1}_{\{x\}} - \mathbb{1}_{\{y\}})$ if x and y lie in the same group G. If $p \nmid \lambda s$, then we get $\mathbb{1}_{\{x\}} - \mathbb{1}_{\{y\}} \in \mathrm{Code}_q(\mathcal{T})^\perp$. Let now

$$\mathcal{C} = \mathrm{Span}_{\mathbb{F}_q}\{\mathbb{1}_{\{x\}} - \mathbb{1}_{\{y\}}, \forall x, y \in X \text{ s.t. } \{x, y\} \subset G \in \mathcal{G}\}$$

We see that $\mathcal{C}^\perp = \{c \in \mathbb{F}_q^{s\ell}, \forall G \in \mathcal{G}, c_{|G} \in \mathrm{Rep}(s)\}$. Therefore we obtain the expected result. □

In the perspective of Problem IV.2, the following section is devoted to the construction of transversal designs with high rate.

V. Explicit constructions of 1-private TD-based PIR protocols

From now on, we denote by $\ell(k)$ the number of servers involved in a given PIR protocol running on a database with k entries, and by $n(k)$ the actual number of symbols stored by all the servers. As it is proved in Theorem IV.1, these two parameters are crucial for the practicality of our PIR schemes, and they respectively correspond to the block size and the number of points of the transversal design used in the construction. In practice, we look for small values of ℓ and n as explained in Problem IV.2.

In this section, we first give two classical instances of transversal designs derived from finite geometries (Subsections V-A and V-B), leading to good PIR parameters. We then show how *orthogonal arrays* produce transversal designs, and we more deeply study a family of such arrays leading to high-rate codes. Subsection V-D is finally devoted to another family of orthogonal arrays whose *divisibility* properties ensure to give an upper bound on the storage overhead of related PIR protocols.

A. Transversal designs from affine geometries

Transversal designs can be built with incidence properties between subspaces of an affine space.

Construction V.1 (Affine transversal design)**.** Let $\mathbb{A}^m(\mathbb{F}_q)$ be the affine space of dimension m over $\mathbb{F}_q$, and $H = \{H_1, \ldots, H_q\}$ be q hyperplanes that partition $\mathbb{A}^m(\mathbb{F}_q)$. We define a transversal design $\mathcal{T}_A(m, q)$ as follows:

- the point set X consists in all the points in $\mathbb{A}^m(\mathbb{F}_q)$;
- the groups in $\mathcal{G}$ are the q hyperplanes from H;
- the blocks in $\mathcal{B}$ are all the 1-dimensional affine subspaces (lines) which do not entirely lie in one of the H_j, $j \in [1, q]$. We also say that such lines are *secant* to the hyperplanes in H.

The design thus defined is a $\mathrm{TD}(q, q^{m-1})$, since an affine line is either contained in one of the H_j, or is 1-secant (*i.e.* has intersection of size 1) to each of them. To complete the study of the parameters of the induced PIR protocol, it remains to compute the dimension of $\mathrm{Code}(\mathcal{T}_A(m, q))$.

Proposition IV.3 first proves that if p does not divide $\lambda s = q$, then the code $\mathrm{Code}_p(\mathcal{T}_A(m, q))$ has poor dimension. Since our goal is to obtain the largest codes as possible, we choose p to be, for instance, the characteristic of the field $\mathbb{F}_q$.

Now notice that all blocks of $\mathcal{T}_A(m, q)$ belong to the block set of the *affine geometry design* $\mathrm{AG}_1(m, q)$ — which is defined as the incidence structure of all points and affine lines in $\mathbb{A}^m(\mathbb{F}_q)$. Thus, the incidence matrix $M_{\mathcal{T}_A(m,q)}$ is a sub-matrix of $M_{\mathrm{AG}_1(m,q)}$, which implies that $\mathrm{Code}_p(\mathrm{AG}_1(m, q)) \subseteq \mathrm{Code}_p(\mathcal{T}_A(m, q))$ for any field $\mathbb{F}_p$. In fact, equality holds as shows the following result.

Proposition V.2. *For every $q = p^e$ and $m \geq 2$, we have*

$$\mathrm{Code}_p(\mathrm{AG}_1(m, q)) = \mathrm{Code}_p(\mathcal{T}_A(m, q))\,.$$

Proof. Denote by $\mathcal{B}^{(\mathrm{AG})}$ the blocks of $\mathrm{AG}_1(m, q)$, and by $\mathcal{B}^{(\mathcal{T})}$ and $\mathcal{G}^{(\mathcal{T})}$ the blocks and groups of $\mathcal{T}_A(m, q)$. Thanks to the previous discussion, we only need to show that for every block $B \in \mathcal{B}^{(\mathrm{AG})}$ contained in a group $G \in \mathcal{G}^{(\mathcal{T})}$, it holds that $\mathbb{1}_B \in \mathrm{Code}_p(\mathcal{T}_A(m, q))^\perp$. For this sake, first notice that $\mathrm{Code}_p(\mathcal{T}_A(m, q))^\perp = \mathrm{Span}\{\mathbb{1}_{B'}, B' \in \mathcal{B}^{(\mathcal{T})}\}$.

Let now $G \in \mathcal{G}^{(\mathcal{T})}$ and $B \in \mathcal{B}^{(\mathrm{AG})}$ such that $B \subseteq G$. Recall that G is a hyperplane of $\mathbb{A}^m(\mathbb{F}_q)$, and let P be a 2-dimensional affine plane of $\mathbb{A}^m(\mathbb{F}_q)$ such that $P \cap G = B$. We claim that $\mathbb{1}_P \in \mathrm{Span}\{\mathbb{1}_{B'}, B' \in \mathcal{B}^{(\mathcal{T})}\}$. Indeed, P admits a partition into affine lines which are secant to every hyperplane in $\mathcal{G}$. Thus $\mathbb{1}_P$ can be written as sum of the characteristic vectors of these lines.

Now let $x \in B$, and $\mathcal{B}_{x,P}^{(\mathcal{T})} \coloneqq \{B' \in \mathcal{B}^{(\mathcal{T})}, x \in B' \subset P\} \subseteq \mathcal{B}^{(\mathcal{T})}$. Define $b^{(x)} = \sum_{B' \in \mathcal{B}_{x,P}^{(\mathcal{T})}} \mathbb{1}_{B'}$. It is clear that $b^{(x)} \in \mathrm{Span}\{\mathbb{1}_{B'}, B' \in \mathcal{B}^{(\mathcal{T})}\}$, and we can notice that

$$\begin{cases} b_x^{(x)} = q = 0, \\ b_i^{(x)} = 0 & \text{for all } i \in B \setminus \{x\}, \\ b_j^{(x)} = 1 & \text{for all } j \in P \setminus B\,. \end{cases}$$

In other words, $b^{(x)} = \mathbb{1}_P - \mathbb{1}_B$, therefore $\mathbb{1}_B \in \mathrm{Span}\{\mathbb{1}_{B'}, B' \in \mathcal{B}^{(\mathcal{T})}\}$. □

The benefit to consider $\mathrm{AG}_1(m, q)$ is that the p-rank of its incidence matrix has been well-studied. For instance, Hamada [12] gives a generic formula to compute the p-rank of a design coming from projective geometry. Yet, as presented in Appendix A, asymptotics are hard to derive from his formula for a generic value of m.

However, if $m = 2$, we know that $\mathrm{rank}_p(\mathrm{AG}_1(2, p^e)) = \binom{p+1}{2}^e$, which implies that

$$\dim(\mathrm{Code}_p(\mathcal{T}_A(2, p^e))) = p^{2e} - \binom{p+1}{2}^e\,.$$

Hence we obtain the following family of PIR protocols.

Proposition V.3. *Let D be a database with $k = p^{2e} - \binom{p+1}{2}^e$ entries, p a prime, $e \geq 1$. There exists a distributed 1-private PIR protocol for D with:*

$$\ell(k) = p^e \quad \textit{and} \quad n(k) = p^{2e}\,.$$

For fixed p and $k \to \infty$, we have

$$\begin{array}{rl} \ell(k) & = \sqrt{k} + \Theta(k^{\frac{1}{2}+c_p}) \quad \textit{and} \\ n(k)/k & = \frac{1}{1-\left(\frac{1+1/p}{2}\right)^e} = 1 + \Theta(k^{c_p}) \to 1\,, \end{array} \tag{1}$$

where $c_p = \frac{1}{2}\log_p(\frac{1+1/p}{2}) < 0$.

Proof. The existence of the PIR protocol is a consequence of the previous discussion, using the family of codes $\mathrm{Code}_p(\mathcal{T}_A(2,p^e))$. Let us state the asymptotics of the parameters. Recall we fix the prime p and we let $e \to \infty$. First we have:

$$\begin{aligned} n(k)/k &= \frac{p^{2e}}{p^{2e} - \binom{p+1}{2}^e} = \frac{1}{1-\left(\frac{1+1/p}{2}\right)^e} \\ &= 1 + \left(\frac{1+1/p}{2}\right)^e + \mathcal{O}\left(\left(\frac{1+1/p}{2}\right)^{2e}\right). \end{aligned} \tag{2}$$

Notice that

$$\begin{aligned} \log_p k &= 2e + \log_p\left(1 - \left(\frac{1+1/p}{2}\right)^e\right) \\ &= 2e + \mathcal{O}\left(\left(\frac{1+1/p}{2}\right)^e\right). \end{aligned}$$

Hence,

$$\begin{aligned} \left(\frac{1+1/p}{2}\right)^e &= \left(\frac{1+1/p}{2}\right)^{\frac{1}{2}\log_p k + \mathcal{O}\left(\left(\frac{1+1/p}{2}\right)^e\right)} \\ &= k^{\frac{1}{2}\log_p\left(\frac{1+1/p}{2}\right)} \times \left(\frac{1+1/p}{2}\right)^{\mathcal{O}\left(\left(\frac{1+1/p}{2}\right)^e\right)} \\ &= \Theta\left(k^{c_p}\right), \end{aligned}$$

since $\left(\frac{1+1/p}{2}\right)^{\mathcal{O}\left(\left(\frac{1+1/p}{2}\right)^e\right)} \to 1$. Using (2) we obtain the asymptotics we claimed on $n(k)/k$.

For $\ell(k)$, we see that $n(k) = \ell(k)^2$. Therefore, we get

$$\ell(k) = \sqrt{k}\sqrt{n(k)/k} = \sqrt{k}\sqrt{1+\Theta(k^{c_p})} = \sqrt{k} + \Theta(k^{\frac{1}{2}+c_p}).$$

□

We give in Table I the dimension of some codes arising from affine transversal designs. Notice that m is not restricted to 2, but we focus on codes with large, since they aimed at being applied in PIR protocols.

Finally, for a better understanding of the parameters we can point out two PIR instances:

- choosing $m = 2$ and $\ell = 4096$, there exists a PIR protocol on a $\simeq 2.0$ MB file with 6 kB of communication and only 3.2% storage overhead;
- for a $\simeq 46$ GB database ($m = 3$, $\ell = 8192$), we obtain a PIR protocol with 39 kB of communication and 27% storage overhead.

m	$\ell = q$	$n = s\ell = q^m$	$k = \dim \mathcal{C}$	$R = k/n$
2	8	64	37	0.578
2	16	256	175	0.684
2	32	1024	781	0.763
2	64	4096	3367	0.822
2	1024	1 048 576	989 527	0.944
2	4096	16 777 216	16 245 775	0.968
2	16 384	268 435 456	263 652 487	0.982
2	65 536	4 294 967 296	4 251 920 575	0.990
3	8	512	139	0.271
3	16	4096	1377	0.336
3	64	262 144	118 873	0.453
3	256	16 777 216	9 263 777	0.552
3	1024	1 073 741 824	680 200 873	0.633
3	8192	549 755 813 888	400 637 408 211	0.729
4	8	4096	406	0.099
4	64	16 777 216	2 717 766	0.162
4	256	4 294 967 296	890 445 921	0.207
5	8	32 768	994	0.030
5	64	1 073 741 824	44 281 594	0.041

TABLE I: Dimension and rate of binary codes $\mathcal{C}$ arising from $\mathcal{T}_A(m,q)$. Remind that the rate R of the code is related to the server storage overhead of the PIR protocol, and that $q = \ell$ is essentially the communication complexity and the number of servers.

B. Transversal designs from projective geometries

The projective space $\mathbb{P}^m(\mathbb{F}_q)$ is defined as $(\mathbb{A}^{m+1}(\mathbb{F}_q) \setminus \{\mathbf{0}\})/\sim$, where for $(\mathbf{P},\mathbf{Q}) \in (\mathbb{A}^{m+1}(\mathbb{F}_q) \setminus \{\mathbf{0}\})^2$, we have $\mathbf{P} \sim \mathbf{Q}$ if and only if there exists $\lambda \in \mathbb{F}_q$ such that $\mathbf{P} = \lambda\mathbf{Q}$. A projective subspace can be defined as the zero set of a collection of linear forms over $\mathbb{F}_q^{m+1}$. In particular, a projective hyperplane is the zero-set of one non-zero linear form over $\mathbb{F}_q^{m+1}$.

Projective geometries are closely related to affine geometries, but contrary to them, there is no partition of the projective space into hyperplanes, since every pair of distinct projective hyperplanes intersects in a projective space of co-dimension 2. To tackle this problem, an idea is to consider the hyperplanes H_i which intersect on a fixed subspace of co-dimension 2 (call it Π_∞). Then, all the sets $H_i \setminus \Pi_\infty$ are disjoint, and their union gives exactly $\mathbb{P}^m(\mathbb{F}_q) \setminus \Pi_\infty$, where $\mathbb{P}^m(\mathbb{F}_q)$ denotes the projective space of dimension m over $\mathbb{F}_q$. Besides, any projective line disjoint from Π_∞ is either contained in one of the H_i, or is 1-secant to all of them. It results to the following construction:

Construction V.4 (Projective transversal design)**.** Let $\mathbb{P}^m(\mathbb{F}_q)$ and Π_∞ defined as above. Let us define

- a point set $X = \mathbb{P}^m(\mathbb{F}_q) \setminus \Pi_\infty$;
- a group set $\mathcal{G} = \{\text{projective hyperplanes } H \subset \mathbb{P}^m(\mathbb{F}_q),\ \Pi_\infty \subset H\}$;
- a block set $\mathcal{B} = \{\text{projective lines } L \subset \mathbb{P}^m(\mathbb{F}_q),\ L \cap \Pi_\infty = \varnothing \text{ and } \forall H \in \mathcal{G},\ L \not\subset H\}$.

Finally, denote by $\mathcal{T}_P(m,q) \coloneqq (X, \mathcal{B}, \mathcal{G})$.

The design $\mathcal{T}_P(m,q)$ is a $\mathrm{TD}(q+1, q^{m-1})$ and, as in the affine setting, its p-rank is related to that of $\mathrm{PG}_1(m,q)$, the classical design of point-line incidences in the projective space $\mathbb{P}^m(\mathbb{F}_q)$. Indeed, the incidence matrix M of $\mathcal{T}_P(m,q)$ is a submatrix of $M_{\mathrm{PG}_1(m,q)}$ from which we removed:

- the columns corresponding to the points in Π_∞,
- the rows corresponding to the lines not in $\mathcal{B}$.

Said differently, the code associated to $\mathcal{T}_P(m,q)$ contains (as a subcode) the Π_∞-shortening of the code associated to $\mathrm{PG}_1(m,q)$. Hence $\dim \mathrm{Code}(\mathcal{T}_P(m,q)) \geq$

$\dim \mathrm{Code}(\mathrm{PG}_1(m,q)) - |\Pi_\infty|$. Contrary to Proposition V.2, we could not prove equality, but this is of little consequence: up to using a subcode of $\mathrm{Code}(\mathcal{T}_P(m,q))$ we can consider PIR protocols on databases with k entries, where $k = \dim \mathrm{Code}(\mathrm{PG}_1(m,q)) - |\Pi_\infty|$.

Once again, for projective geometries Hamada's formula gets simpler for $m = 2$, and leads to the following proposition.

Proposition V.5. *Let D be a database with $k = p^{2e} + p^e - \binom{p+1}{2}^e - 1$ entries, p a prime and $e \geq 1$. There exists a distributed 1-private PIR protocol for D with:*

$$\ell(k) = p^e + 1 \quad \text{and} \quad n(k) = p^{2e} + p^e \,.$$

Asymptotics are the same as in Equation (1).

In order to emphasize that the two previous constructions are asymptotically the same, we draw the rates of the codes involved in these two kinds of PIR schemes in Figure 4.

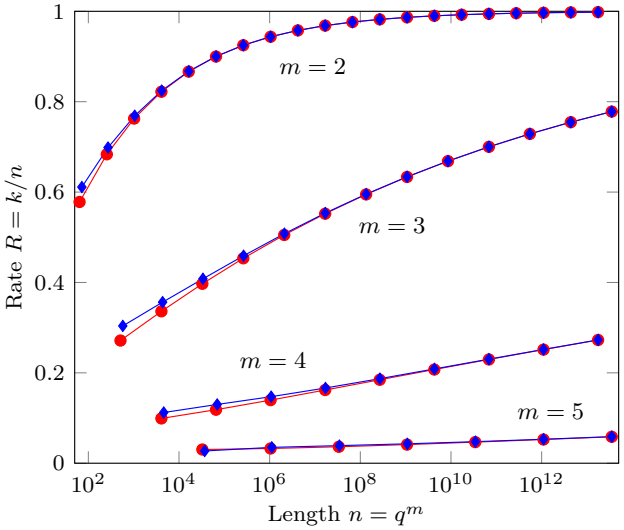

Fig. 4: Rate of binary codes coming from $\mathcal{T}_A(m,q)$ (in red) and $\mathcal{T}_P(m,q)$ (in blue). For every fixed m, we let q grow.

C. Orthogonal arrays and the incidence code construction

In this subsection, we first recall a way to produce plenty of transversal designs from other combinatorial constructions called *orthogonal arrays*.

Definition V.6 (orthogonal array)**.** Let $\lambda, s \geq 1$ and $\ell \geq t \geq 1$, and let A be an array with ℓ columns and λs^t rows, whose entries are elements of a set S of size s. We say that A is an orthogonal array $\mathrm{OA}_\lambda(t,\ell,s)$ if, in any subarray A' of A formed by t columns and all its rows, every row vector from S^t appears exactly λ times in the rows of A'. We call λ the *index* of the orthogonal array, t its *strength* and ℓ its *degree*. If t (resp. λ) is omitted, it is understood to be 2 (resp. 1). If both these parameters are omitted we write $A = \mathrm{OA}(\ell, s)$.

From now on, for convenience we restrict Definition V.6 to orthogonal arrays with no repeated column and no repeated row. Next paragraph introduces a link between orthogonal arrays and transversal designs.

1) Construction of transversal designs from orthogonal arrays: We can build a transversal design $\mathrm{TD}(\ell, s)$ from an orthogonal array $\mathrm{OA}(\ell, s)$ with the following construction, given as a remark in [8, ch.II.2].

Construction V.7 (Transversal designs from orthogonal arrays)**.** Let A be an $\mathrm{OA}(\ell, s)$ of strength $t = 2$ and index $\lambda = 1$ with symbol set S, $|S| = s$, and denote by $\mathrm{Rows}(A)$ the s^2 rows of A. We define the point set $X = S \times [1, \ell]$. To each row $c \in \mathrm{Rows}(A)$ we associate a block

$$B_c \coloneqq \{(c_i, i), i \in [1, \ell]\} \,,$$

so that the block set is defined as

$$\mathcal{B} \coloneqq \{B_c, c \in \mathrm{Rows}(A)\} \,.$$

Finally, let $\mathcal{G} \coloneqq \{S \times \{i\}, i \in [1, \ell]\}$. Then $(X, \mathcal{B}, \mathcal{G})$ is a transversal design $\mathrm{TD}(\ell, s)$.

Example V.8. A very simple example of this construction is given in Figure 5, where for clarity we use letters for elements of the symbol set $\{a, b\}$, while the columns are indexed by integers. On the left-hand side, A is an $\mathrm{OA}_1(2, 3, 2)$ with symbol set $\{a, b\}$. On the right-hand side, the associated transversal design $\mathrm{TD}(3, 2)$ is represented as a hypergraph: the nodes are the points of the design, the "columns" of the graph form the groups, and a block consists in all nodes linked with a path of a fixed color. One can check that every pair of nodes either belongs to the same group or is linked with one path.

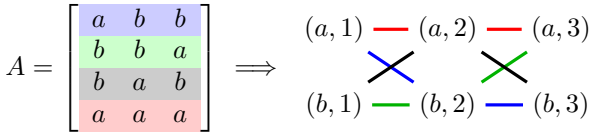

Fig. 5: A representation of the construction of a transversal design from an orthogonal array.

Remark V.9. Listed in rows, all the codewords of a (generic) code $\mathcal{C}_0$ give rise to an orthogonal array, whose strength t is derived from the dual distance d' of $\mathcal{C}_0$ by $t = d' - 1$. Notice that for linear codes, the dual distance is simply the minimum distance of the dual code, but it can also be defined for non-linear codes (see [15, Ch.5.§5.]). More details about the link between orthogonal arrays and codes can also be found in [8]. For example, the orthogonal array of Figure 5 comes from the binary parity-check code of length 3 (by replacing a by 0 and b by 1). One can check that its dual distance is 3 and its associated transversal design has strength 2.

Given a code $\mathcal{C}_0$, we denote by $A_{\mathcal{C}_0}$ the orthogonal array it defines (see Remark V.9) and by $\mathcal{T}_{\mathcal{C}_0}$ the transversal design built from $A_{\mathcal{C}_0}$ thanks to Construction V.7.

Example V.10. Let $\mathbf{x} = (x_1, \ldots, x_\ell)$ be an ℓ-tuple of pairwise distinct elements of $\mathbb{F}_q$ and denote by $\mathrm{RS}_2(\mathbf{x})$ the Reed-Solomon code of length ℓ and dimension 2 over $\mathbb{F}_q$ with evaluation points $\mathbf{x}$:

$$\mathrm{RS}_2(\mathbf{x}) \coloneqq \{(f(x_1), \ldots, f(x_\ell)),\ f \in \mathbb{F}_q[X],\ \deg f < 2\} \,.$$

Then, $\mathrm{RS}_2(\mathbf{x})$ has dual distance 3, so its codewords form an orthogonal array $A_{\mathrm{RS}_2(\mathbf{x})} = \mathrm{OA}(\ell, q)$ of strength 2. Now, one can use Construction V.7 to obtain a transversal design $\mathcal{T}_{\mathrm{RS}_2(\mathbf{x})} = \mathrm{TD}(\ell, q)$. The point set is $X = \mathbb{F}_q \times [1, \ell]$, and the blocks are "labeled Reed-Solomon codewords", that is, sets of the form $\{(c_i, i),\ i \in [1, \ell]\}$ with $c \in \mathrm{RS}_2(\mathbf{x})$. The ℓ groups correspond to the ℓ coordinates of the code: $G_i = \mathbb{F}_q \times \{i\}$, $1 \leq i \leq \ell$.

We can finally sum up our construction by introducing the code $\mathrm{Code}_q(\mathcal{T}_{\mathcal{C}_0})$ arising from the transversal design defined by $\mathcal{C}_0$. To the best of our knowledge, the construction $\mathcal{C}_0 \mapsto \mathrm{Code}_q(\mathcal{T}_{\mathcal{C}_0})$ is new. We name $\mathrm{Code}_q(\mathcal{T}_{\mathcal{C}_0})$ the *incidence code* of $\mathcal{C}_0$, since its parity-check matrix $M_{\mathcal{T}_{\mathcal{C}_0}}$ essentially stores incidence relations between all the codewords in $\mathcal{C}_0$.

Definition V.11 (incidence code)**.** Let $\mathcal{C}_0$ be a (generic) code of length ℓ over an alphabet S of size s. The *incidence code* of $\mathcal{C}_0$ over $\mathbb{F}_q$, denoted $\mathrm{IC}_q(\mathcal{C}_0)$, is the $\mathbb{F}_q$-linear code of length $n = s\ell$ built from the transversal design $\mathcal{T}_{\mathcal{C}_0}$, that is:

$$\mathrm{IC}_q(\mathcal{C}_0) := \mathrm{Code}(\mathcal{T}_{\mathcal{C}_0})\,.$$

Notice that the field $\mathbb{F}_q$ does not need to be the alphabet S of the code $\mathcal{C}_0$.

Incidence codes are introduced in order to design PIR protocols, as summarizes Figure 6. We can show that, if $\mathcal{C}_0$ has dual distance more than 3, then the induced PIR protocol is 1-private. A generalisation is formally proved in Corollary VI.8.

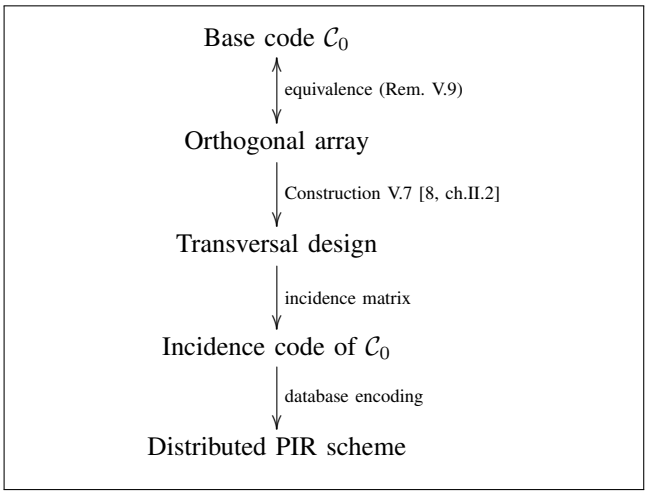

Fig. 6: A distributed PIR scheme using the incidence code construction.

Example V.12. Here we provide a full example of the construction of an incidence code. Let $\mathcal{C}_0$ be the full-length Reed-Solomon code of dimension 2 over the field $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2 = \alpha + 1\}$. The orthogonal array associated to $\mathcal{C}_0$ is composed by the following list of codewords:

$$A = \begin{pmatrix} 0, & 0, & 0, & 0 \\ 1, & 1, & 1, & 1 \\ \alpha, & \alpha, & \alpha, & \alpha \\ \alpha^2, & \alpha^2, & \alpha^2, & \alpha^2 \\ 0, & 1, & \alpha, & \alpha^2 \\ 0, & \alpha, & \alpha^2, & 1 \\ 0, & \alpha^2, & 1, & \alpha \\ 1, & 0, & \alpha^2, & \alpha \\ 1, & \alpha^2, & \alpha, & 0 \\ 1, & \alpha, & 0, & \alpha^2 \\ \alpha, & \alpha^2, & 0, & 1 \\ \alpha, & 0, & 1, & \alpha^2 \\ \alpha, & 1, & \alpha^2, & 0 \\ \alpha^2, & \alpha, & 1, & 0 \\ \alpha^2, & 1, & 0, & \alpha \\ \alpha^2, & 0, & \alpha, & 1 \end{pmatrix}$$

Using Construction V.7, we get a transversal design $\mathcal{T}_{\mathcal{C}_0} = (X, \mathcal{B}, \mathcal{G})$ with 16 points (4 groups made of 4 points) and 16 blocks. Let us recall how we map a row of A to a word in $\{0,1\}^{16}$. For instance, consider the fifth row:

$$a := A_5 = (0, 1, \alpha, \alpha^2)\,.$$

We turn a into a block $B_a := \{(0,1), (1,2), (\alpha,3), (\alpha,4)\} \in \mathcal{B}$, and we build the incidence vector $\mathbb{1}_{B_a}$ of the block B_a over the point set $X = \{(\beta, i), i \in [1,4], \beta \in \mathbb{F}_4\}$. Of course, in order to see $\mathbb{1}_{B_a}$ as a word in $\{0,1\}^{16}$, we need to order elements in X, for instance:

$$\begin{pmatrix} (0,1), (1,1), (\alpha,1), (\alpha^2,1), \\ (0,2), (1,2), (\alpha,2), (\alpha^2,2), \\ (0,3), (1,3), (\alpha,3), (\alpha^2,3), \\ (1,4), (1,4), (\alpha,4), (\alpha^2,4) \end{pmatrix}.$$

Using this ordering, we get:

$$\mathbb{1}_{B_a} = \big(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1\big) \in \{0,1\}^{16}\,.$$

By computing all the $\mathbb{1}_{B_a}$ for $a \in \mathrm{Rows}(A)$, we obtain the incidence matrix M of the transversal design $\mathcal{T}_{\mathcal{C}_0}$:

$$M = \left(\begin{array}{c:c:c:c} 1000 & 1000 & 1000 & 1000 \\ 0100 & 0100 & 0100 & 0100 \\ 0010 & 0010 & 0010 & 0010 \\ 0001 & 0001 & 0001 & 0001 \\ 1000 & 0100 & 0010 & 0001 \\ 1000 & 0010 & 0001 & 0100 \\ 1000 & 0001 & 0100 & 0010 \\ 0100 & 1000 & 0001 & 0010 \\ 0100 & 0001 & 0010 & 1000 \\ 0100 & 0010 & 1000 & 0001 \\ 0010 & 0001 & 1000 & 0100 \\ 0010 & 1000 & 0100 & 0001 \\ 0010 & 0100 & 0001 & 1000 \\ 0001 & 0010 & 0100 & 1000 \\ 0001 & 0100 & 1000 & 0010 \\ 0001 & 1000 & 0010 & 0100 \end{array}\right),$$

Notice that this matrix can be quickly obtained by respectively replacing entries 0, 1, α and α^2 in the array A by the binary 4-tuples (1000), (0100), (0010) and (0001) in the matrix

M (of course this map depends on the ordering of X we have chosen, but another choice would lead to a column-permutation-equivalent matrix, hence a permutation-equivalent code). Notice that in matrix M, coordinates lying in the same group of the transversal design have been distinguished by dashed vertical lines.

Matrix M then defines, over any extension $\mathbb{F}_{2^e}$ of the prime field $\mathbb{F}_2$, the dual code of the so-called incidence code $\mathrm{IC}_{2^e}(\mathcal{C}_0)$. For all values of e, the incidence codes $\mathrm{IC}_{2^e}(\mathcal{C}_0)$ have the same generator matrix of 2-rank 7, being:

$$G = \left(\begin{array}{c|c|c|c} 1001 & 0000 & 0011 & 1010 \\ 0101 & 0000 & 0110 & 0011 \\ 0011 & 0000 & 0101 & 0110 \\ 0000 & 1001 & 0101 & 1100 \\ 0000 & 0101 & 0011 & 0110 \\ 0000 & 0011 & 0110 & 0101 \\ 0000 & 0000 & 1111 & 1111 \end{array}\right).$$

2) A deeper analysis of incidence codes coming from linear MDS codes of dimension 2: Incidence codes lead to an innumerably large family of PIR protocols — as many as there exists codes $\mathcal{C}_0$ — but most of them are not practical for PIR protocols (essentially because the kernel of the incidence matrix is too small). To simplify their study, one can first remark that intuitively, the more blocks a transversal design, the larger its incidence matrix, and consequently, the lower the dimension of its associated code. But the number of blocks of $\mathcal{T}_{\mathcal{C}_0}$ is the cardinality of $\mathcal{C}_0$. Hence, informally the smaller the code $\mathcal{C}_0$, the larger $\mathrm{IC}(\mathcal{C}_0)$.

We recall that a $[n,k,d]$ linear code is said to be maximum distance separable (MDS) if it reaches the Singleton bound $n+1=k+d$. Besides, the dual code of an MDS code is also MDS, hence its dual distance is $k+1$. In this paragraph we analyse the incidence codes constructed with MDS codes of dimension 2. Their interest lies in being the smallest codes with dual distance 3, which is the minimal setting for defining 1-private PIR protocols.

Generalized Reed-Solomon codes are the best-known family of MDS codes.

Definition V.13 (generalized Reed-Solomon code)**.** Let $\ell \geq k \geq 1$. Let also $\mathbf{x} \in \mathbb{F}_q^\ell$ be a tuple of pairwise distinct so-called *evaluation points*, and $\mathbf{y} \in (\mathbb{F}_q^\times)^\ell$ be the *column multipliers*. We associate to $\mathbf{x}$ and $\mathbf{y}$ the *generalized Reed-Solomon* (GRS) code:

$$\mathrm{GRS}_k(\mathbf{x},\mathbf{y}) \coloneqq \{(y_1 f(x_1),\dots,y_\ell f(x_\ell)), \\ f \in \mathbb{F}_q[X], \deg f < k\}.$$

Generalized Reed-Solomon codes $\mathrm{GRS}_k(\mathbf{x},\mathbf{y})$ are linear MDS codes of dimension k over $\mathbb{F}_q$, and they give usual Reed-Solomon codes when $\mathbf{y}=(1,\dots,1)$. Moreover, GRS codes are essentially the only MDS codes of dimension 2, as states the following lemma whose proof can be found in the Appendix.

Lemma V.14. *All $[\ell,2,\ell-1]$ MDS codes over $\mathbb{F}_q$ with $2 \leq \ell \leq q$ are generalized Reed-Solomon codes.*

Let us study the consequences of Lemma V.14 in terms of transversal designs. We say a map $\phi : X \to X'$ is an isomorphism between transversal designs $(X,\mathcal{B},\mathcal{G})$ and $(X',\mathcal{B}',\mathcal{G}')$ if it is one-to-one and if it preserves the incidence relations, or in other words, if ϕ is invertible on the points, blocks and groups:

$$\phi(X)=X', \quad \phi(\mathcal{B})=\mathcal{B}', \quad \phi(\mathcal{G})=\mathcal{G}'.$$

Lemma V.15. *Let $\mathcal{C},\mathcal{C}'$ be two codes such that $\mathcal{C}' = \mathbf{y} * \mathcal{C}$ for some $\mathbf{y} \in (\mathbb{F}_q^\times)^\ell$, where $*$ is the coordinate-wise product of ℓ-tuples. Recall $\mathcal{T}_\mathcal{C}$, $\mathcal{T}_{\mathcal{C}'}$ are the transversal designs they respectively define. Then, $\mathcal{T}_\mathcal{C}$ and $\mathcal{T}_{\mathcal{C}'}$ are isomorphic.*

Proof. Write $\mathcal{T}_\mathcal{C} = (X,\mathcal{B},\mathcal{G})$ and $\mathcal{T}_{\mathcal{C}'} = (X',\mathcal{B}',\mathcal{G}')$. From the definition it is clear that $X = X' = \mathbb{F}_q \times [1,\ell]$ and $\mathcal{G} = \mathcal{G}' = \{\mathbb{F}_q \times \{i\}, 1 \leq i \leq \ell\}$. Now consider the blocks sets. We see that $\mathcal{B} = \{\{(c_i,i), 1 \leq i \leq \ell\}, c \in \mathcal{C}\}$ and $\mathcal{B}' = \{\{(y_i c_i, i), 1 \leq i \leq \ell\}, c \in \mathcal{C}\}$. Let:

$$\begin{array}{rccc} \phi_\mathbf{y}: & \mathbb{F}_q \times [1,\ell] & \to & \mathbb{F}_q \times [1,\ell] \\ & (x,i) & \mapsto & (y_i x, i) \end{array}$$

The vector $\mathbf{y}$ is $*$-invertible, hence $\phi_\mathbf{y}$ is one-to-one on the point set X. It remains to notice that $\phi_\mathbf{y}$ maps $\mathcal{G}$ to itself since it only acts on the first coordinate, and that $\phi_\mathbf{y}(\mathcal{B})$ is exactly $\mathcal{B}'$ by definition of $\mathcal{C}$ and $\mathcal{C}'$. □

Proposition V.16. *Let $2 \leq \ell \leq q$ and $\mathcal{C}_0$ be an $[\ell,2,\ell-1]_q$ linear (MDS) code. Let also $\mathbb{F}_p$ be any finite field. The incidence code $\mathrm{IC}_p(\mathcal{C}_0)$ is permutation-equivalent to $\mathrm{IC}_p(\mathrm{RS}_2(\mathbf{x}))$, with $\mathbf{x} \in \mathbb{F}_q^\ell$, $x_i \neq x_j$.*

Proof. Lemma V.14 shows that all $[\ell,2,\ell-1]_q$ linear codes $\mathcal{C}_0$ can be written as $\mathbf{y} * \mathrm{RS}_2(\mathbf{x})$ for some $\mathbf{x} \in \mathbb{F}_q^\ell$. Moreover, with the previous notation $\phi_\mathbf{y}(\mathcal{T}_{\mathrm{RS}_2(\mathbf{x})}) = \mathcal{T}_{\mathbf{y}*\mathrm{RS}_2(\mathbf{x})}$, so we have $u \in \mathrm{IC}_p(\mathbf{y} * \mathrm{RS}_2(\mathbf{x}))$ if and only if $u \in \mathrm{Code}_p(\phi_\mathbf{y}(\mathcal{T}_{\mathrm{RS}_2(\mathbf{x})}))$. Now, let:

$$\begin{array}{rccc} \tilde{\phi}_\mathbf{y}: & \mathbb{F}_p^X & \to & \mathbb{F}_p^X \\ & u=(u_x)_{x\in X} & \mapsto & (u_{\phi_\mathbf{y}(x)})_{x\in X} \end{array}.$$

Clearly $\tilde{\phi}_\mathbf{y}(\mathrm{IC}_p(\mathrm{RS}_2(\mathbf{x}))) = \mathrm{Code}_p(\phi_\mathbf{y}(\mathcal{T}_{\mathrm{RS}_2(\mathbf{x})}))$ and $\tilde{\phi}_\mathbf{y}$ is a permutation of coordinates. So $\mathrm{IC}_p(\mathcal{C}_0)$ is permutation-equivalent to $\mathrm{IC}_p(\mathrm{RS}_2(\mathbf{x}))$ which proves the result. □

In our study of incidence codes of 2-dimensional MDS codes $\mathcal{C}_0$, the previous proposition allows us to restrict our work on Reed-Solomon codes $\mathcal{C}_0 = \mathrm{RS}_2(\mathbf{x})$ with $\mathbf{x}$ an ℓ-tuple on pairwise distinct $\mathbb{F}_q$-elements.

A first result proves that if $\mathbf{x}$ contains all the elements in $\mathbb{F}_q$, then $\mathrm{IC}_q(\mathrm{RS}_2(\mathbf{x}))$ is the code which has been previously studied in subsection V-A. More precisely,

Proposition V.17. *The following two codes are equal up to permutation:*

1) *$\mathcal{C}_1 = \mathrm{IC}_q(\mathrm{RS}_2(\mathbb{F}_q))$, the incidence code over $\mathbb{F}_q$ of the full-length Reed-Solomon code of dimension 2 over $\mathbb{F}_q$;*
2) *$\mathcal{C}_2$, the code over $\mathbb{F}_q$ based on the transversal design $\mathcal{T}_A(2,q)$.*

Proof. It is sufficient to show that the transversal design defined by $\mathcal{C}_0 = \mathrm{RS}_2(\mathbb{F}_q)$ is isomorphic to $\mathcal{T}_A(2,q)$. Let us

enumerate $\mathbb{F}_q = \{x_1, \ldots, x_q\}$. We recall that $\mathcal{T}_{\mathcal{C}_0} = (X, \mathcal{B}, \mathcal{G})$ where:

$$\begin{aligned} X &= \mathbb{F}_q \times [1, q], \\ \mathcal{B} &= \{\{(c_i, i), i \in [1, q]\}, c \in \mathcal{C}_0\}, \\ \mathcal{G} &= \{\{(\alpha, i), \alpha \in \mathbb{F}_q\}, i \in [1, q]\}, \end{aligned}$$

and that $\mathcal{T}_A(2, q) = (X', \mathcal{B}', \mathcal{G}')$ with:

$$\begin{aligned} X' &= \mathbb{F}_q \times \mathbb{F}_q, \\ \mathcal{B}' &= \{\{(ax_i + b, x_i), i \in [1, q]\}, (a, b) \in \mathbb{F}_q^2\} \\ \mathcal{G}' &= \{\{(\alpha, x_i), \alpha \in \mathbb{F}_q\}, i \in [1, q]\}. \end{aligned}$$

In the light of the above, one defines $\phi : X \to X'$, $(\alpha, i) \mapsto (\alpha, x_i)$, which is clearly one-to-one and satisfies $\phi(\mathcal{G}) = \mathcal{G}'$. Moreover, a codeword $c \in \mathcal{C}_0$ is the evaluation of a polynomial of degree ≤ 1 over $\mathbb{F}_q$. Hence for some $(a, b) \in \mathbb{F}_q^2$, we have $c_i = ax_i + b, \forall i$. This proves that ϕ extends to a one-to-one map $\mathcal{B} \to \mathcal{B}'$, giving the desired isomorphism. □

It remains to study the case of tuples $\mathbf{x}$ of length $\ell < q$. First, one may notice that $\mathrm{IC}_q(\mathrm{RS}_2(\mathbf{x}))$ is a shortening of $\mathrm{IC}_q(\mathrm{RS}_2(\mathbb{F}_q))$. Indeed, we have the following property:

Lemma V.18. *Let $\mathcal{C}_0$ be a linear code of length ℓ over $\mathbb{F}_q$, and $\overline{\mathcal{C}_0}$ be a puncturing of $\mathcal{C}_0$ on s positions. Then for all prime powers q', $\mathrm{IC}_{q'}(\overline{\mathcal{C}_0})$ is a shortening of $\mathrm{IC}_{q'}(\mathcal{C}_0)$ on the coordinates corresponding to s groups of the transversal design $\mathcal{T}_{\mathcal{C}_0}$.*

Proof. Without loss of generality, we can assume that $\mathcal{C}_0$ is punctured on its s last coordinates in order to give $\overline{\mathcal{C}_0}$. Let us analyse the link between $\mathcal{T}_{\overline{\mathcal{C}_0}} = (\overline{X}, \overline{\mathcal{B}}, \overline{\mathcal{G}})$ and $\mathcal{T}_{\mathcal{C}_0} = (X, \mathcal{B}, \mathcal{G})$. We have:

$$\begin{array}{lll} \overline{X} & = \mathbb{F}_q \times [1, \ell - s] & \subset X, \\ \overline{\mathcal{G}} & = \{\mathbb{F}_q \times \{i\},\ i \in [1, \ell - s]\} & \subset \mathcal{G}, \\ \overline{\mathcal{B}} & = \{B \cap \overline{X},\ B \in \mathcal{B}\} & \end{array}$$

Let $\mathcal{C} = \mathrm{IC}_{q'}(\mathcal{C}_0)$ and $\overline{\mathcal{C}} = \mathrm{IC}_{q'}(\overline{\mathcal{C}_0})$. For clarity, we index words in $\mathcal{C}$ (resp. $\overline{\mathcal{C}}$) by X (resp. $\overline{X}$). For $\overline{c} \in \mathbb{F}_{q'}^{\overline{X}}$, we define $\mathrm{ext}(\overline{c}) := c \in \mathbb{F}_{q'}^{X}$, such that $c_{|\overline{X}} = \overline{c}$ and $c_{|X \setminus \overline{X}} = 0$. By definition of code's puncturing/shortening, all we need to prove is:

$$\overline{\mathcal{C}} = \{\overline{c} \in \mathbb{F}_{q'}^{\overline{X}}, \mathrm{ext}(\overline{c}) \in \mathcal{C}\}.$$

Remind that $\overline{\mathcal{C}}$ is defined as the set of $\overline{c} \in \mathbb{F}_{q'}^{\overline{X}}$ satisfying $\sum_{b \in \overline{B}} \overline{c}_b = 0$ for every $\overline{B} \in \overline{\mathcal{B}}$. Hence we have:

$$\begin{aligned} \overline{c} \in \overline{\mathcal{C}} &\iff \sum_{b \in \overline{B}} \overline{c}_b = 0, \quad \forall \overline{B} \in \overline{\mathcal{B}} \\ &\iff \sum_{b \in B \cap \overline{X}} \overline{c}_b = 0, \quad \forall B \in \mathcal{B} \\ &\iff \sum_{b \in B \cap \overline{X}} \mathrm{ext}(\overline{c})_b \\ &\qquad + \sum_{b \in B \cap (X \setminus \overline{X})} \mathrm{ext}(\overline{c})_b = 0, \quad \forall B \in \mathcal{B} \\ &\iff \sum_{b \in B} \mathrm{ext}(\overline{c})_b = 0, \quad \forall B \in \mathcal{B} \\ &\iff \mathrm{ext}(\overline{c}) \in \mathcal{C} \end{aligned}$$

We conclude the proof by pointing out that $X \setminus \overline{X}$ is a union of s distinct groups from $\mathcal{G}$. □

Despite this result, incidence codes of Reed-Solomon codes $\mathrm{RS}_2(\mathbf{x})$ remain hard to classify for $|\mathbf{x}| = \ell < q$. Indeed, for a given length $\ell < q$, some $\mathrm{IC}(\mathrm{RS}(\mathbf{x}))$ appear to be non-equivalent. Their dimension can even be different, as shows an exhaustive search on $\mathrm{IC}_{16}(\mathrm{RS}(\mathbf{x}))$ with pairwise distinct $\mathbf{x} \in \mathbb{F}_q^\ell$, $q = 16$ and $\ell = 5$: we observe that 48 of these codes have dimension 24 while the 4320 others have dimension 22. Further interesting research would then be to understand the values of $\mathbf{x}$ leading to the largest codes, for a fixed length $|\mathbf{x}| = \ell$.

D. High-rate incidence codes from divisible codes

In this subsection, we prove that linear codes $\mathcal{C}_0$ satisfying a *divisibility* condition yield incidence codes whose rate is roughly greater than $1/2$. Let us first define divisible codes.

Definition V.19 (divisibility of a code)**.** Let $p \geq 2$. A linear code is p-divisible if p divides the Hamming weight of all its codewords.

A study of the incidence matrix which defines an incidence code leads to the following property.

Lemma V.20. *Let $\mathcal{C}_0$ be a code of length ℓ over a set S, and let $\mathcal{T}$ be the transversal design associated to $\mathcal{C}_0$. We denote by M the incidence matrix of $\mathcal{T}$, where rows of M are indexed by codewords from $\mathcal{C}_0$. Then we have:*

$$(MM^T)_{c,c'} = \ell - d(c, c') \quad \forall c, c' \in \mathcal{C}_0,$$

where $d(\cdot, \cdot)$ denotes the Hamming distance.

Proof. For clarity we adopt the notation $M[c, (\alpha, i)]$ for the entry of M which is indexed by the codeword $c \in \mathcal{C}_0$ (for the row), and $(\alpha, i) \in S \times [1, \ell]$ (for the column). We also denote by $\mathbb{1}_{\mathcal{U}(c,i,\alpha)} \in \{0, 1\}$ the boolean value of the property $\mathcal{U}$, that is, $\mathbb{1}_{\mathcal{U}(c,i,\alpha)} = 1$ if and only if $\mathcal{U}(c, i, \alpha)$ is satisfied. Now, let $c, c' \in \mathcal{C}_0$.

$$\begin{aligned} (MM^T)_{c,c'} &= \sum_{\alpha \in S,\, i \in [1,\ell]} M[c, (\alpha, i)] M[c', (\alpha, i)] \\ &= \sum_{\alpha \in S,\, i \in [1,\ell]} \mathbb{1}_{c_i = \alpha} \mathbb{1}_{c'_i = \alpha} \\ &= \sum_{i=1}^{\ell} \sum_{\alpha \in S} \mathbb{1}_{c_i = c'_i = \alpha} \\ &= \sum_{i=1}^{\ell} \mathbb{1}_{c_i = c'_i} \\ &= \ell - d(c, c'). \end{aligned}$$

□

Hence, if some prime p divides ℓ as well as the weight of all the codewords in $\mathcal{C}_0$, then the product MM^T vanishes over any extension of $\mathbb{F}_p$, and M is a parity-check matrix of a code containing its dual. A more general setting is analyzed in the following proposition.

Proposition V.21. *Let $\mathcal{C}_0$ be a linear code of length ℓ over S, $|S| = s$. Let also $\mathcal{C} = \mathrm{IC}_q(\mathcal{C}_0)$ with $\mathrm{char}(\mathbb{F}_q) = p$. Denote the length of $\mathcal{C}$ by $n = \ell s$. If $\mathcal{C}_0$ is p-divisible, then*

$$\mathcal{C}^\perp \cap \mathcal{C}_{par} \subseteq \mathcal{C}\,,$$

where $\mathcal{C}_{par}$ denotes the parity-check code of length n over $\mathbb{F}_q$. In particular, we get $\dim \mathcal{C} \geq \frac{n-1}{2}$.

Moreover, if $p \mid \ell$, then $\mathcal{C}^\perp \subseteq \mathcal{C}$ and $\dim \mathcal{C} \geq \frac{n}{2}$.

Proof. Let M be the incidence matrix of the transversal design $\mathcal{T}_{\mathcal{C}_0}$. Also denote by J and J' the all-ones matrices of respective size $|\mathcal{C}_0| \times n$ and $|\mathcal{C}_0| \times |\mathcal{C}_0|$. If we assume that $\mathcal{C}_0$ is p-divisible, then Lemma V.20 translates into

$$MM^T = \ell J' \mod p \tag{3}$$

while an easy computation shows that

$$MJ^T = \ell J'\,.$$

Hence, over $\mathbb{F}_q$ we obtain

$$M(M-J)^T = 0 \tag{4}$$

which brings us to consider the code A of length n generated over $\mathbb{F}_q$ by the matrix $M - J$. Equation (4) indicates that $A \subseteq \mathcal{C}$. Let $\mathcal{C}_{\text{par}} \coloneqq \{c \in \mathbb{F}_q^n, \sum_i c_i = 0\}$ be the parity-check code of length n over $\mathbb{F}_q$. Notice that $c \in \mathcal{C}_{\text{par}} \iff cJ^T = 0$ and $uJ = 0 \iff uJ' = 0$. If $p \nmid \ell$, this leads to:

$$\begin{aligned}
\mathcal{C}^\perp \cap \mathcal{C}_{\text{par}} &= \{c = uM \in \mathbb{F}_q^n, cJ^T = 0\} \\
&= \{c = uM \in \mathbb{F}_q^n, \ell u J' = 0\} \\
&= \{c = uM \in \mathbb{F}_q^n, uJ = 0\} \\
&= \{u(M-J) \in \mathbb{F}_q^n, uJ = 0\} \subseteq A \subseteq \mathcal{C}\,.
\end{aligned}$$

On the other hand, if $p \mid \ell$, then equation (3) turns into $MM^T = 0$, meaning that $\mathcal{C}^\perp \subseteq \mathcal{C}$.

Finally, the first bound on the dimension comes from

$$\dim \mathcal{C} \geq \dim(\mathcal{C}^\perp \cap \mathcal{C}_{\text{par}}) \geq \dim \mathcal{C}^\perp - 1 = n - \dim \mathcal{C} - 1\,,$$

while the second one is straightforward. □

In terms of PIR protocols, previous result translates into the following corollary.

Corollary V.22. *Let p be a prime, and assume there exists a p-divisible linear code of length ℓ_0 over $\mathbb{F}_q$. Then, there exists $k \geq (\ell_0 q - 1)/2$ such that we can build a distributed PIR protocol for a k-entries database over $\mathbb{F}_q$, and whose parameters are $\ell(k) = \ell_0$ and $n(k) = \ell_0 q \leq 2k + 1$.*

Divisible codes over small fields have been well-studied, and contain for instance the extended Golay codes [15, ch.II.6], or the famous MDS codes of dimension 3 and length $q + 2$ over $\mathbb{F}_q$ [15, ch.XI.6].

Example V.23. The extended binary Golay code is a self-dual $[24, 12, 8]_2$ linear code. It produces a transversal design with 24 groups, each storing 2 points. Its associated incidence code $\mathrm{Code}_2(\mathrm{Golay})$ has length $n = 24 \times 2 = 48$ and dimension ≥ 24, and by computation we can show that this bound is tight.

Remark V.24. In our application for PIR protocols, we would like to find divisible codes $\mathcal{C}_0$ defined over large alphabets (compared to the code length), but these two constraints seem to be inconsistent. For instance, the binary Golay code presented in Example V.23 leads to a PIR protocol with a too expensive communication cost (24 bits of communication for an original file of size... 24 bits: that is exactly the communication cost of the trivial PIR protocol where the whole database is downloaded). Nevertheless, Example V.23 represents the worst possible case for our construction, in a sense that the rate of $\mathrm{IC}_2(\mathrm{Golay}_2)$ is exactly $1/2$ (it attains the lower bound), and that each server stores 2 bits (which is the smallest possible). Codes with better rate and/or with larger server storage capability would then give PIR protocols with relevant communication complexity. For instance, the extended ternary Golay code gives better parameters — see Example VI.9.

Divisible codes over large fields seems not to have been thoroughly studied (to the best of our knowledge), since coding theorists use to consider codes over small alphabets as more practical. We hope that our construction of PIR protocols based on divisible codes may encourage research in this direction.

VI. PIR protocols with better privacy

When servers are colluding, the PIR protocol based on a simple transversal design does not ensure a sufficient privacy, because the knowledge of two points on a block gives some information on it. To solve this issue, we propose to use orthogonal arrays with higher strength t.

A. Generic construction and analysis

In the previous section, classical ($t = 2$) orthogonal arrays were used to build transversal designs. Considering higher values of t, we naturally generalize the latter as follows:

Definition VI.1 (t-transversal designs)**.** Let $\ell \geq t \geq 1$. A t-transversal design is a block design $\mathcal{D} = (X, \mathcal{B})$ equipped with a group set $\mathcal{G} = \{G_1, \ldots, G_\ell\}$ partitioning X such that:

- $|X| = s\ell$;
- any group has size s and any block has size ℓ;
- for any $T \subseteq [1, \ell]$ with $|T| = t$ and for any $(x_1, \ldots, x_t) \in G_{T_1} \times \ldots \times G_{T_t}$, there exist exactly λ blocks $B \in \mathcal{B}$ such that $\{x_1, \ldots, x_t\} \subset B$.

A t-transversal design with parameters s, ℓ, t, λ is denoted $t\text{-TD}_\lambda(\ell, s)$, or $t\text{-TD}(\ell, s)$ if $\lambda = 1$.

Given a t-transversal design $\mathcal{T}$, we can build a $(t-1)$-private PIR protocol with the exactly the same steps as in section IV. First, we define the code $\mathcal{C} = \mathrm{Code}_q(\mathcal{T})$ associated to the design according to Definition III.7, and then we follow the algorithm given in Figure 2. Since a t-transversal design is also a 2-transversal design for $t \geq 2$, the analysis is identical for every PIR feature, except for the security where it remains very similar.

Security ($(t-1)$-privacy). Let T be a collusion of servers of size $|T| \leq t-1$. For varying $i \in I$, the distributions $\mathcal{Q}(i)_{|T}$ are the same because there are exactly $\lambda s^{t-1-|T|} \geq \lambda \neq 0$

blocks which contain both i and the queries known by the servers in T.

To sum up, the following theorem holds:

Theorem VI.2. *Let D be a database with k entries over $\mathbb{F}_q$, and $\mathcal{T} = t\text{-TD}(\ell, s)$ be a t-transversal design, whose incidence matrix has rank $\ell s - k$ over $\mathbb{F}_q$. Then, there exists an ℓ-server $(t-1)$-private PIR protocol with:*

- *only 1 symbol to read for each server,*
- *$\ell - 1$ field operations for the user,*
- *$\ell \log(sq)$ bits of communication,*
- *a (total) storage overhead of $(\ell s - k)\log q$ bits on the servers.*

B. Instances and results

1) t-transversal designs from curves of degree $\leq t-1$: Looking for instances of t-transversal designs, it is natural to try to generalise the transversal designs of Construction V.1. An idea is to turn affine lines into higher degree curves.

Construction VI.3. Let X be the set of points in the affine plane $\mathbb{F}_q^2$, and $\mathcal{G} = \{G_1, \dots, G_q\}$ be a partition of X in q parallel lines. W.l.o.g. we choose the following partition: $G_i = \{(y, \alpha_i), y \in \mathbb{F}_q\}$ for each $\alpha_i \in \mathbb{F}_q$. Blocks are now defined as the sets of the form

$$B_F = \{(F(x), x), x \in \mathbb{F}_q\}, \text{ where } F \in \mathbb{F}_q[x], \deg F \leq t-1.$$

Lemma VI.4. *The design $(X, \mathcal{B}, \mathcal{G})$ given in Construction VI.3 forms a t-transversal design $t\text{-TD}_1(q, q)$.*

Proof. The group set indeed partitions X into q groups, each of size q. It remains to check the incidence property. Let $\{G_{T_1}, \dots, G_{T_t}\}$ be a set of t distinct groups, and let $((y_{T_1}, x_{T_1}), \dots, (y_{T_t}, x_{T_t})) \in G_{T_1} \times \dots \times G_{T_t}$. From Lagrange interpolation theorem, we know there exists a unique polynomial $F \in \mathbb{F}_q[X]$ of degree $\leq t-1$ such that:

$$F(x_{T_j}) = y_{T_j} \quad \forall 1 \leq j \leq t\,.$$

Said differently, there is a unique block which contains the t points $\{(y_{T_j}, x_{T_j})\}_{1 \leq j \leq t}$. □

We do not yet analyse the rank properties of these designs, since Construction VI.3 corresponds to a particular case of the generic construction given below.

2) t-transversal designs from orthogonal arrays of strength t: In this paragraph we give a generic construction of t-transversal designs, which is a simple generalisation of the way we build transversal designs with orthogonal arrays (Subsection V-C).

Construction VI.5. Let A be an orthogonal array $\text{OA}_\lambda(t, \ell, s)$ on a symbol set S. Recall that the array A is composed of rows $a_i = (a_{i,j})_{1 \leq j \leq \ell}$ for $1 \leq i \leq \lambda s^t$. We define the following design:

- its point set is $X = S \times [1, \ell]$;
- its group set is $\mathcal{G} = \{S \times \{i\}, 1 \leq i \leq \ell\}$;
- its blocks are $B_i = \{(a_{i,j}, j), 1 \leq j \leq \ell\}$ for all $a_i \in \text{Rows}(A)$.

Proposition VI.6. *If A is an $\text{OA}_\lambda(t, \ell, s)$, then the design defined with A by Construction VI.5 is a $t\text{-TD}_\lambda(\ell, s)$.*

Proof. It is clear that $\mathcal{G}$ is a partition of X and that blocks and groups have the claimed size. Now focus on the incidence property. Let $T \subset [1, \ell]$ with $|T| = t$, and let $(x_1, \dots, x_t) \in G_{T_1} \times \dots \times G_{T_t}$. We need to prove that there are exactly λ blocks $B \in \mathcal{B}$ such that $\{x_1, \dots, x_t\} \subset B$.

Consider the map from blocks in $\mathcal{B}$ to rows of A given by:

$$\begin{array}{rccc} \psi : & \mathcal{B} & \to & \text{Rows}(A) \\ & B_i = \{(a_{i,j}, j), 1 \leq j \leq \ell\} & \mapsto & (a_{i,1}, \dots, a_{i,\ell}) \end{array}$$

Since we assumed that orthogonal arrays have no repeated row, the map ψ is one-to-one. Denote by $x' = (x'_1, , \dots, x'_t) \in S^t$ the vector formed by the first coordinates of $(x_1, , \dots, x_t) \in X^t$. From the definition of an orthogonal array of strength t and index λ, we know that x' appears exactly λ times in the submatrix of A defined by the columns indexed by T. Hence this defines λ preimages in $\mathcal{B}$, which proves the result. □

Remark VI.7. As we noticed before, Construction VI.3 is a particular case of Construction VI.5. Indeed, a block $B_F = \{(F(x), x), x \in \mathbb{F}_q\}$, with $\deg F \leq t-1$ is in one-to-one correspondence with a codeword c_F of a Reed-Solomon code of dimension t.

Corollary VI.8. *Let $\mathcal{C}_0$ be a code of length ℓ and dual distance $t+2 \leq \ell$ over a set S of size s. Then, $\text{IC}_q(\mathcal{C}_0)$ defines a t-private PIR protocol.*

Proof. Let A be the orthogonal array defined by $\mathcal{C}_0$. We know that A has strength $t+1$ (see *e.g.* [15]), hence from Proposition VI.6, the associated transversal design is a $(t+1)\text{-TD}(\ell, s)$. Theorem VI.2 then ensures that the PIR protocol induced by this transversal design is t-private. □

As in Section V, if the code $\mathcal{C}_0$ is divisible, then we can give a lower bound on the rate of its incidence code. We provide two examples in finite (and small) length.

Example VI.9. A first example would be to consider extended Golay codes. Indeed, they are known to be divisible by their characteristic [15, ch.II.6], they have large dual distance, and Proposition V.21 then ensures their incidence codes have non-trivial rate. In Remark V.24, we noticed that the binary Golay code does not lead to a practical PIR protocol due to a large communication complexity. Thus, let us instead consider the $[12, 6, 6]_3$ extended ternary Golay code, that we denote Golay_3. It is self-dual, hence $d^\perp(\text{Golay}_3) = 6$. Then, $\mathcal{C} = \text{IC}_{3^e}(\text{Golay}_3)$, $e \geq 1$, has length 36 and Proposition V.21 shows that $\dim \mathcal{C} \geq 18$ (the bound can be proved to be tight by computation). Hence, the associated PIR protocol works on a raw file of 18 $\mathbb{F}_{3^e}$-symbols encoded into 36, uses 12 servers (each storing 3 $\mathbb{F}_{3^e}$-symbols) and resists any collusion of one third (*i.e.* 4) of them.

Example VI.10. A second example arises from the exceptional $[q+2, 3, q]_q$ MDS codes in characteristic 2 [15, ch.XI.6]. For instance, for $q = 4$, we obtain a 2-private PIR protocol with 6 servers, each storing 4 symbols of $\mathbb{F}_{2^e}$ for some $e \geq 1$. Once again, the dimension of the incidence code attains the lower bound, here $k = 12$.

Example VI.11. Examples of incidence codes which do not attain the lower bound of Proposition V.21 come from binary Reed-Muller codes of order 1, denoted $\mathrm{RM}_2(m,1)$. These codes are 2-divisible since they are known to be equivalent to extended Hamming codes. They also have length $n = 2^m$ and dual distance $d^\perp = n/2$.

For instance, $\mathrm{RM}_2(3,1)$ provides an incidence code of dimension $k = 11 > 8$, that is, a 2-private 8-server PIR protocol on a database with 11 $\mathbb{F}_{2^e}$-symbols, where each server stores 2 symbols. For $m = 4$, $\mathrm{RM}_2(4,1)$ gives a 6-private 16-server PIR protocol on a database with 20 $\mathbb{F}_{2^e}$-symbols, each server storing 2 symbols. We conjecture that $\mathrm{IC}_2(\mathrm{RM}_2(m,1))$ leads to a $(2^{m-1}-2)$-private 2^m-server PIR protocol on a database with 2^m+m symbols, each server storing 2 symbols.

As pointed out in Subsection V-C, high-rate incidence codes $\mathcal{C} = \mathrm{IC}(\mathcal{C}_0)$ have the best chance to occur when the dimension of $\mathcal{C}_0$ is small, since the cardinality of $\mathcal{C}_0$ is the number of rows in a (non-full-rank) parity-check matrix which defines $\mathcal{C}$. Besides, in order to define t-private PIR protocols, we need an orthogonal array of strength $t+2$, *i.e.* a code $\mathcal{C}_0$ with dual distance $t+2$. Conciliating both constraints, we are tempted to pick MDS codes of dimension $t+1$.

A well-known family of MDS codes is the family of Reed-Solomon codes. For $\mathcal{C}_0 = \mathrm{RS}_{t+1}(\mathbb{F}_q)$ and varying values of q and t, we were able to compute the rate of $\mathrm{IC}(\mathcal{C}_0)$, and these codes lead to t-private PIR protocols with communication complexity approximately $\sqrt{n}$, where n is the length of the encoded database. These rates are presented in Figure 7 and as expected, the rate of our families of incidence codes decreases with t, the privacy parameter. Figure 7 also shows that Reed-Solomon-based instances cannot expect to reach at the same time constant information rate and resistance to a constant fraction of colluding servers.

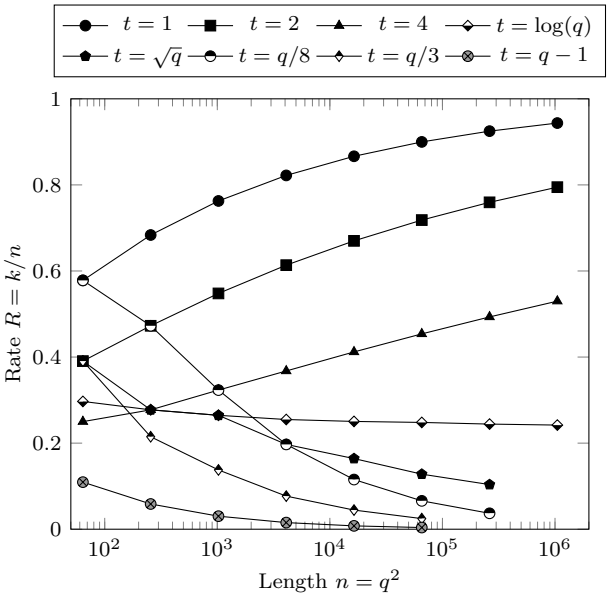

Fig. 7: Rate of incidence codes of $\mathcal{C}_0$ that are used for building t-private PIR protocols. Codes $\mathcal{C}_0$ are full-length Reed-Solomon codes of dimension $t+1$ (dual distance $t+2$) over $\mathbb{F}_q$. Associated PIR protocols then need q servers, each storing q symbols.

VII. Comparison with other works

Our construction fits into the model of distributed (or coded) PIR protocols, which is currently instantiated in a few schemes, notably the construction of Augot *et al.* [2] and all the works involving the use of PIR codes initiated by Fazeli *et al* [11]. We recall that we aimed at building PIR protocols with very low burden for the servers, in terms of storage *and* computation. While PIR codes are a very efficient way to reduce the storage overhead, they do not cut down the computation complexity of the original replication-based PIR protocol used for the emulation.

Hence, for the sake of consistency, we will only compare the parameters of our PIR schemes with those of the multiplicity code construction presented in [2].

Sketch of the construction [2]. Multiplicity codes $\mathcal{C}$ have the property that a codeword $c \in \mathcal{C}$ can be seen as the vector of evaluations of a multivariate polynomial $f_c \in \mathbb{F}_q[X_1,\ldots,X_m]$ and its derivatives over the space $\mathbb{F}_q^m$, where $\mathbb{F}_q$ denotes the finite field with q elements. Every affine line of the space $\mathbb{F}_q^m$ then induces linear relations between f_c and its derivatives, which translates into low-weight parity-check equations for the codewords. This allows to define a local decoder for $\mathcal{C}$: when trying to retrieve a symbol D_i indexed by $i \in \mathbb{F}_q^m$, one can pick random affine lines going through i and recover D_i by computing short linear combinations of the symbols associated to the evaluations of f_c and their derivatives along these lines. We refer to [14] for more details on these codes.

Augot *et al.* [2] realized that partitioning $\mathbb{F}_q^m$ into q parallel hyperplanes gives rise to storage improvements. By splitting the encoded database according to these hyperplanes and giving one part to each of the q servers, they obtained a huge cut on both the total storage and the number of servers, while keeping an acceptable communication complexity. Their construction requires a minor modification of the LDC-based PIR protocol of Figure 1; indeed, in the query generation process, the only server which holds the desired symbol must receive a random query. Nevertheless, the PIR scheme they built was at that time the only one to let the servers store less than twice the size of the database. Moreover, the precomputation of the encoding of the database ensures an optimal computational complexity for the servers. As noticed previously, we emphasize the significance of this feature when the database is very frequently queried.

Parameters of the distributed PIR protocol [2] based on multiplicity codes. This PIR scheme depends on four main parameters: the field size q, the dimension m of the underlying affine space, the multiplicity order s and the maximal degree d of evaluated polynomials. For an error-free and collusion-free setting, $d = s(q-1)-1$ is the optimal choice. Let $\sigma = \binom{m+s-1}{m}$. The associated PIR protocol uses q servers to store an original database containing $\binom{d+m}{m}$ $\mathbb{F}_q$-symbols, but encoded into codewords of length q^m, where each symbol has size $\sigma \log q$ bits. Hence the redundancy (in bits) of the

scheme is:

$$\rho = \left(\sigma q^m - \binom{s(q-1)+m-1}{m}\right) \log q\,.$$

Concerning the communication complexity, let us only focus on the download cost (which is often the bottleneck in practice). The multiplicity code local decoding algorithm needs to query symbols of σ distinct lines of the space. Hence, each server must answer σ symbols of size $\sigma \log q$ bits. Thus it leads to a download communication complexity of

$$\gamma = \sigma^2 q \log q \quad \text{bits.}$$

Note about the comparison strategy. We consider a database D of size 100 MB. Since the protocols may be initially constructed for databases of smaller size k, we split D into k chunks of size $|D|/k$. Hence, when running the PIR protocol, the user is allowed to retrieve a whole chunk, and the chunk size will be precised in our tables. For instance, in the first row of Table II, one shall understand that the user is able to retrieve 31.1kB of the database privately, with 1.99MB of communication, while each server only produces 1 operation over the chunks (of size 31.1kB) it holds.

Tables II and III first reveal that our PIR schemes are more storage efficient than the PIR schemes relying on multiplicity codes. Moreover, our constructions provide a better *communication rate* (defined as the ratio between communication cost and chunk size), though the multiplicity code PIR protocols allow to retrieve smaller chunks (hence is more flexible).

Remark VII.1. Recent constructions of PIR protocols (for instance results of Sun and Jafar such that [17]) lead to better parameters in terms of communication complexity. However, we once more emphasize that we aimed at minimizing the computation carried out by the servers, which is a feature that is mostly not considered in those works.

VIII. Conclusion

In this work, we have presented a generic construction of codes yielding distributed PIR protocols with optimal server computational complexity, in a sense that each server only has to read one symbol of the part of the database it stores. Our construction makes use of transversal designs, whose incidence properties ensure a natural distribution of the coded database on the servers, as well as the privacy of the queries. Our PIR protocols also feature efficient reconstructing steps since the user has to compute a simple linear combination of the symbols it receives. Finally, they require low storage for the servers and acceptable communication complexity.

We instantiated our construction with classical transversal designs coming from affine and projective geometries, and with transversal designs emerging from orthogonal arrays of strength 2. The last construction that we call *incidence code* can even be generalized, since stronger orthogonal arrays lead to PIR protocols with a better resilience to collusions.

The generality of our construction allows the user to choose appropriate settings according to the context (low storage capability, few colluding servers, *etc.*). It also raises the question of finding transversal designs with the most practical PIR parameters for a given context. Indeed, while affine and projective geometries give excellent PIR parameters for the servers (low computation, low storage), there seems to remain room for improving the communication complexity and the number of needed servers.

References

[1] Edward F. Assmus and Jennifer D. Key. *Designs and Their Codes*. Cambridge Tracts in Mathematics. Cambridge University Press, 1992.

[2] Daniel Augot, Françoise Levy-dit-Vehel, and Abdullatif Shikfa. A Storage-Efficient and Robust Private Information Retrieval Scheme Allowing Few Servers. In Dimitris Gritzalis, Aggelos Kiayias, and Ioannis G. Askoxylakis, editors, *Cryptology and Network Security - 13th International Conference, CANS 2014, Heraklion, Crete, Greece, October 22-24, 2014. Proceedings*, volume 8813 of *Lecture Notes in Computer Science*, pages 222–239. Springer, 2014.

[3] Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Jean-François Raymond. Breaking the $O(n^{1/(2k-1)})$ Barrier for Information-Theoretic Private Information Retrieval. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 261–270. IEEE Computer Society, 2002.

[4] Amos Beimel, Yuval Ishai, and Tal Malkin. Reducing the Servers' Computation in Private Information Retrieval: PIR with Preprocessing. *J. Cryptology*, 17(2):125–151, 2004.

[5] Benny Chor and Niv Gilboa. Computationally Private Information Retrieval. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 304–313. ACM, 1997.

[6] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private Information Retrieval. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, 23-25 October 1995*, pages 41–50. IEEE Computer Society, 1995.

[7] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private Information Retrieval. *J. ACM*, 45(6):965–981, 1998.

[8] Charles J. Colbourn and Jeffrey H. Dinitz. *Handbook of Combinatorial Designs, Second Edition*. Chapman & Hall/CRC, 2006.

[9] Zeev Dvir and Sivakanth Gopi. 2-Server PIR with Subpolynomial Communication. *J. ACM*, 63(4):39:1–39:15, 2016.

[10] Klim Efremenko. 3-Query Locally Decodable Codes of Subexponential Length. *SIAM J. Comput.*, 41(6):1694–1703, 2012.

[11] Arman Fazeli, Alexander Vardy, and Eitan Yaakobi. Codes for Distributed PIR with Low Storage Overhead. In *IEEE International Symposium on Information Theory, ISIT 2015, Hong Kong, China, June 14-19, 2015*, pages 2852–2856. IEEE, 2015.

Instance	download communication	complexity (#op./server)	storage overhead	chunk size
$\mathcal{T}_A(m=2, q=64)$	1.99 MB	1	22.7 MB	31.1 kB
$\mathcal{T}_A(m=3, q=64)$	56 kB	1	126 MB	882 B
$\mathrm{Mult}(q=64, m=3, s=6)$	2.32 MB	56	64.8 MB	16 B
$\mathrm{Mult}(q=64, m=4, s=2)$	15 kB	5	694 MB	13 B

TABLE II: Comparison of some distributed PIR protocols with 64 servers on a 100MB initial database. Parameters of multiplicity codes have been chosen in order to obtain simultaneously low communication complexity and storage overhead.

Instance	download communication	complexity (#op./server)	storage overhead	chunk size
$\mathcal{T}_A(m=2, q=8)$	22.7 MB	1	76 MB	2.83 MB
$\mathcal{T}_A(m=3, q=8)$	6.03 MB	1	281 MB	754 kB
$\mathrm{Mult}(q=8, m=4, s=2)$	8.8 MB	5	797 MB	117 kB
$\mathrm{Mult}(q=8, m=6, s=3)$	2.86 MB	28	3.24 GB	1.2 kB

TABLE III: Comparison of some distributed PIR protocols with 8 servers on a 100MB initial database. We notice that for $q=8$ servers, there only exist a few non-Reed-Muller ($s \geq 2$) multiplicity codes whose associated PIR protocols have communication complexity strictly less than the size of the original database.

[12] Noboru Hamada. The rank of the incidence matrix of points and d-flats in finite geometries. *Journal of Science of the Hiroshima University, Series A-I (Mathematics)*, 32(2):381–396, 1968.

[13] Jonathan Katz and Luca Trevisan. On the Efficiency of Local Decoding Procedures for Error-Correcting Codes. In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 80–86. ACM, 2000.

[14] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-Rate Codes with Sublinear-Time Decoding. *J. ACM*, 61(5):28:1–28:20, 2014.

[15] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. North Holland, 1977.

[16] Douglas R. Stinson. *Combinatorial Designs – Constructions and Analysis*. Springer, 2004.

[17] Hua Sun and Syed Ali Jafar. The Capacity of Private Information Retrieval. *IEEE Trans. Information Theory*, 63(7):4075–4088, 2017.

[18] Razan Tajeddine and Salim El Rouayheb. Private Information Retrieval from MDS Coded Data in Distributed Storage Systems. In *IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, July 10-15, 2016*, pages 1411–1415. IEEE, 2016.

[19] Sergey Yekhanin. Towards 3-query Locally Decodable Codes of Subexponential Length. *J. ACM*, 55(1):1:1–1:16, 2008.

[20] Sergey Yekhanin. Locally Decodable Codes. *Foundations and Trends in Theoretical Computer Science*, 6(3):139–255, 2012.

APPENDIX

A. Hamada's formula

Hamada [12] gives a generic formula to compute the p-rank of a projective geometry design $\mathrm{PG}_t(m,q)$, for $q=p^e$:

$$\mathrm{rank}_p(\mathrm{PG}_t(m,q)) = \sum_{(s_0,\ldots,s_e)\in S} \prod_{j=0}^{e-1} \sum_{i=0}^{L(s_{j+1},s_j)} (-1)^i \binom{m+1}{i} \binom{m+s_{j+1}p-s_j-ip}{m}$$

where $S \subset \mathbb{Z}^{e+1}$ contains elements $(s_0,\ldots,s_e)$ such that:

$$\begin{cases} s_0 = s_e \\ t+1 \leq s_j \leq m+1 \\ 0 \leq s_{j+1}p - s_j \leq (m+1)(p-1)\,, \end{cases}$$

and $L(s_{j+1},s_j) = \lfloor \frac{s_{j+1}p-s_j}{p} \rfloor$.

The p-rank of the associated affine geometry design $\mathrm{AG}_t(m,q)$ can be derived from the projective one by:

$$\mathrm{rank}_p(\mathrm{AG}_t(m,q)) = \mathrm{rank}_p(\mathrm{PG}_t(m,q)) - \mathrm{rank}_p(\mathrm{PG}_t(m-1,q))\,.$$

Despite its heavy expression, Hamada's formula can be simplified by picking very specific values of m, p or e. For instance we have:

$$\begin{aligned} m=2: &\quad \forall p,e, \mathrm{rank}_p\,\mathrm{AG}_1(2,p^e) = \tbinom{p+1}{2}^e\,, \\ e=1: &\quad \forall p,m, \mathrm{rank}_p\,\mathrm{AG}_1(m,p) = p^m - \tbinom{m+p-2}{m}\,. \end{aligned}$$

For $(m,e)=(3,2)$, we get

$$\forall p, \quad \mathrm{rank}_p\,\mathrm{AG}_1(3,p^2) = \left(p^3 - \tbinom{p+1}{3}\right)^2 + 2\tbinom{p}{2}\tbinom{p+1}{3}\,,$$

this equality being found by interpolation, since $\mathrm{rank}_p(\mathrm{AG}_1(m,p^e))$ is a polynomial of degree at most me in p.

B. Proof of Lemma V.14

Let us recall the result we want to state.

Lemma. *All $[\ell, 2, \ell-1]$ MDS codes over $\mathbb{F}_q$ with $2 \leq \ell \leq q$ are generalized Reed-Solomon codes.*

Proof. First we know that GRS codes are MDS.

Let $\mathcal{C}$ be an $[\ell, 2, \ell-1]_q$ code with $2 \leq \ell \leq q$. Since $\mathcal{C}$ is MDS, it has dual distance $d^\perp = 3$, and we claim there exists a codeword $c \in \mathcal{C}$ with Hamming weight ℓ. Indeed, let $G = (P_1, \ldots, P_\ell)$ be a generator matrix of $\mathcal{C}$, where $P_i \in \mathbb{F}_q^2$ is written in column. Notice that each point P_i is non-zero (otherwise $d^\perp = 1$) and $0, P_i, P_j$ are not on the same line for $i \neq j$ (otherwise $d^\perp = 2$). Moreover codewords in $\mathcal{C}$ are simply evaluations of bilinear maps $\mu : \mathbb{F}_q^2 \to \mathbb{F}_q$ over $(P_1, \ldots, P_\ell)$:

$$\mathcal{C} = \{(\mu(P_1), \ldots, \mu(P_\ell)), \mu \in \mathcal{L}(\mathbb{F}_q^2, \mathbb{F}_q)\},$$

and the P_i's are not all on the same line (otherwise, $\dim \mathcal{C} \leq 1$).

Since $\ell \leq q$, there exists $Q = (Q_0, Q_1) \in \mathbb{F}_q^2 \setminus \{0\}$ such that Q does not lie in the (vector) line defined by any of the P_i's. Let now $\mu_Q(X, Y) = Q_1 X - Q_0 Y$: it is a non-zero bilinear form which must vanish on a line of $\mathbb{F}_q^2$, and since $\mu_Q(Q) = 0$, it vanishes on the one spanned by Q. To sum up, for every $i \in [1, \ell]$, we have $\mu_Q(P_i) \neq 0$. Hence, $c = (\mu_Q(P_1), \ldots, \mu_Q(P_\ell))$ belongs to $\mathcal{C}$ and has Hamming weight ℓ.

Let now $u \in \mathcal{C}$ such that $\{c, u\}$ spans $\mathcal{C}$. We denote by $c * u$ the coordinate-wise product $(c_1 u_1, \ldots, c_\ell u_\ell)$ and by $\mathbf{1}$ the all-one vector of length ℓ. Then $c = \mathbf{1} * c$ and $u = c * (c^{-1} * u)$, where c^{-1} is the coordinate-wise inverse of c through $*$. Hence, the code $\mathcal{C}$ can be written $c * \mathcal{C}'$ where $\mathcal{C}'$ has $G' = \begin{pmatrix} \mathbf{1} \\ c^{-1} * u \end{pmatrix}$ as generator matrix. It means that $\mathcal{C}$ is the GRS code with evaluation points $\mathbf{x} = c^{-1} * u$, multipliers $\mathbf{y} = c$ and dimension 2. □

ACKNOWLEDGMENTS

The author would like to thank Françoise Levy-dit-Vehel and Daniel Augot for their valuable comments, and more specifically the first collaborator for her helpful guidance all along the writing of the paper.